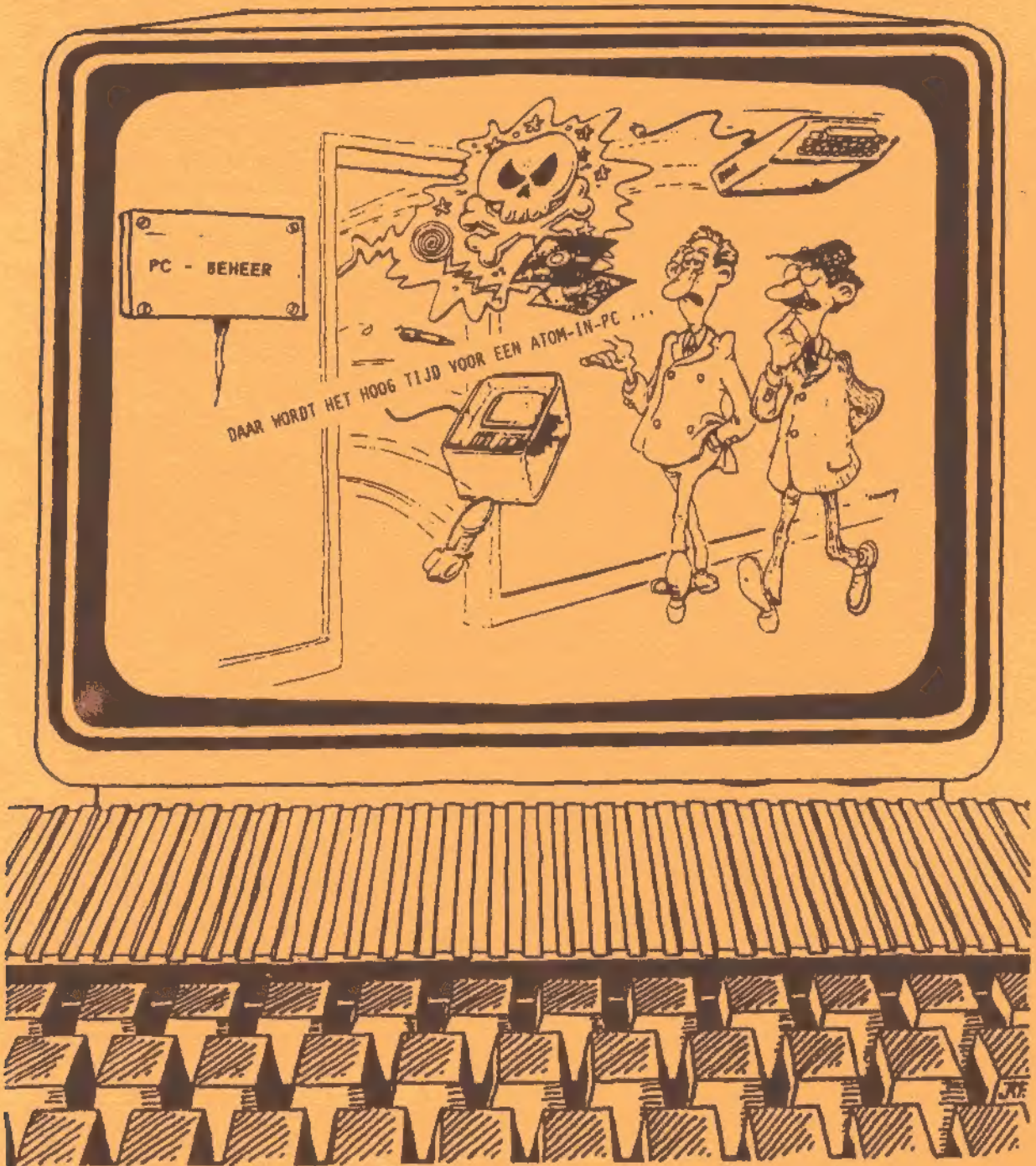


ATOM NIEUWS

JAARGANG : 13
NUMMER : 3



FEDERATIE VAN ATOMCLUBS NEDERLAND - BELGIE.

Voorzitter :	Secretaris:	Penningmeester:
-----	-----	-----
P.v.Kuik	J.Hartog	T.Rutten
Zuideinde 54-a	Keyenbergseweg 60	Berkenlaan 24
1843 JP Groot-Schermer	6871 WK Renkum	3737 RN Groenekan
tel. 02997-1902	tel. 08373-13757	tel.03461-3495

Contributie 1994 : fl. 25,00 : Atom Computerclub : Giro 5244293.

Redactie Atom Nieuws	Redactieadres A.N.	Ledenadministratie
-----	-----	-----
B.Tossaint 043-431675	B.Tossaint	T.Rutten
W.Truijen 09-3289564792	Fatimaplein 85	Berkenlaan 24
R.Leurs 046-370650	6214 TW Maastricht	3737 RN Groenekan
	tel. 043-431675	tel. 03461-3495

UITERSTE DATUM INLEVERING KOPY VOOR NR. 13-4 : 1 DEC 1994

Clubwinkel	ATOM-BULLETTIN-BORD speciaal v. ATOM-in-PC	
-----	-----	
J.Hartog	R.Bronsdijk	: Unicorn BBS (H.Derksen)
Keyenbergseweg 60		: 085-425506 in de gebieden
6871 WK Renkum	inloggen op	: ATOMFILE en ATOMMESSAGE de
tel. 08373-13757	tel.020-6512861	: laatste versies ATOM-PC -
		systemsoftware .

De Clubwinkel :

80-koloms-video-kaart excl. onderdelen	fl.	5,00
Combikaart 91 versie 1 : zie SPS-Printservice		
Z-80-kaart voor CP/M , exclusief onderdelen	fl.	10,00
ACORN NIEUWS 1982, 97 pagina's samenvatting	fl.	1,00
ATOM NIEUWS jaargang 1983 , +/- 450 pag.	fl.	1,00
ATOM NIEUWS Jaargang 1984 , +/- 650 pag.	fl.	1,00
ATOM NIEUWS Jaargang 1985 , +/- 650 pag.	fl.	1,00
ATOM NIEUWS Jaargang 1986 , +/- 500 pag.	fl.	1,00
ATOM NIEUWS Jaargang 1987 , +/- 300 pag.	fl.	1,00
ATOM-WARE : deel 1 : Atom-basic interpreter , 98 pag.	fl.	1,00
ATOM-WARE : deel 2 : Atom-disk operating syst.68 pag.	fl.	1,00
ATOM-WARE : deel 3 : Monitor operating system 80 pag.	fl.	1,00

Levering geschiedt via uw regionale penningmeester, of rechtstreeks, via de penningmeester van de federatie . Bij rechtstreekse bestelling stort U het bedrag van het gewenste artikel , vermeerderd met fl. 4,00 portokosten , op de giro van de federatie , met de vermelding van de naam van het artikel en uw lidmaatschapsnummer.

I N H O U D S O P G A V E

Pag.	Titel	Schrijver
2	Uit de federatie	
3	Inhoudsopgave	
3	Atomlanddag	
4	Van de redactie	
5	STOP	R.Leurs
6 - 16	I/O aan de ATOM-in-PC	R.Leurs e.a.
11	Regio-nieuws	
17 - 21	ATOM-TALK	J.Geene
22 - 31	Een LCD-display aan de Atom	G.Baltus
32 - 36	WIN-ATOM , stand van zaken	L.Bijnagte
36	Mededelingen van de regio Den Haag	Th. Waayer
37 - 39	Koppeling Electron en PC	R.Leurs
40 - 43	Het programmeren van de 65802	L.Bijnagte
44	Files zoeken in een directory	R.Leurs
45 - 48	Versienummers laten zien	W.Truyen
49 - 50	CGA-monitor aansluiten aan Electron	R.Leurs
51 - 57	Zullen we een lijntje trekken	L.Bijnagte
58 - 59	Intikken en rennen maar	R.Leurs
60	Regioadressen	

=====

[]

[]

A.C.C. LANDDAG NA-JAAR 1994

=====

[]

=====

DATUM : 8 OCTOBER 1994 ,van ca. 10.00 tot ca.17.00 u.

PLAATS : PAROCHIECENTRUM ,Melkweg 5,DE BILT.

Op de A27 de afslag Utrecht-Oost,maar let vooral op de kleine witte aanduiding =VEEMARKT=, de Biltse Rading afrijden tot de splitsing Groenekan/Bilthoven ,hier r.a. ,20 meter verder l.a.

Voor busreizigers : Vanaf Utrecht CS lijn 57,stopt op de Melkweg
Breng brood mee, voor koffie wordt gezorgd.

Voorwaarde : Iedereen is zelf verantwoordelijk voor zijn eigen
apparatuur ,etc.

- DEMO's: o.a.* R.Leurs : ATOM IN PC, verbeterde versie
- * idem : I/O KAART VOOR ATOM IN PC
- * J.Geene : Nieuwe ontwikkelingen op monitor-
gebied en ATOM-TALK
- * G.Baltus : LCD aan ATOM (niet zeker)

Het Bestuur van de Federatie.

=====

VOORWOORD VAN DE REDACTIE

Aangezien Pascalle en ik beiden lid zijn van de Atom Club krijgen wij dus steeds twee exemplaren van Atom Nieuws. Daardoor was het mogelijk om het volgende experiment uit te voeren:

Aangezien ik mijn eigen artikelen al gelezen heb, scheurde ik ze uit de laatste twee nummers van Atom Nieuws (nrs 1 en 2 van 1994). Ook de redactionele en federatieve mededelingen haalde ik er uit. Helaas bleven er toen nog slechts 35 pagina's tekst over van andere auteurs. Precies drie andere namen. De laatste twee nummers waren geschreven door vier personen! Dit is geen goede ontwikkeling. Zo gaat Atom Nieuws ten gronde.

Ik kan me niet indenken dat we binnen de hele club met slechts vier mensen nog actief bezig zijn. Op de landdagen, die toch meestal door meer dan twintig mensen bezocht worden, zie ik steeds nieuwe ontwikkelingen, zowel hardware als software. Echter in Atom Nieuws zoek ik steeds vergeefs naar deze onderwerpen. Dus volg het voorbeeld van Guido, Leendert en Lotje en laat het de redactie weten als u ergens mee bezig bent dat ook voor anderen interessant kan zijn. Heeft u geen tijd om een artikel te schrijven? Wij wel! En we willen ook voor u een verhaal schrijven, maar dan moeten we het wel even weten ...

Ook dit nummer hebben we hoofdzakelijk weer volgestopt met artikelen van mij. Ik vind er bijna niets meer aan als Atom Nieuws in de bus valt. Ik ken alle verhalen al. Zo kan ik nu al vertellen dat de Electron aan de PC gekoppeld is en dat de I/O kaart voor de Atom bijna gereed is. Overigens werkt een CGA monitor ook goed aan een Electron. Tenslotte doe ik een poging om de aardige rubriek 'Intikken en runnen maar' weer terug te halen. Maar daarvoor hebben we wel uw bijdrage nodig. Voor de eerst volgende keer hebben we zelfs een prijs uitgelooft voor de beste inzending.

Leden, zet eens wat op papier, stuur dit naar de redactie en voorkom zo dat we aan het eind van het jaar alleen maar een accept giro kaart moeten rondsturen. Ik zie het in gedachte al de begeleidende brief:

Geachte leden,

I.v.m. gebrek aan kopij ontvangt u hierbij een accept giro om uw contributie voor 1995 te voldoen.

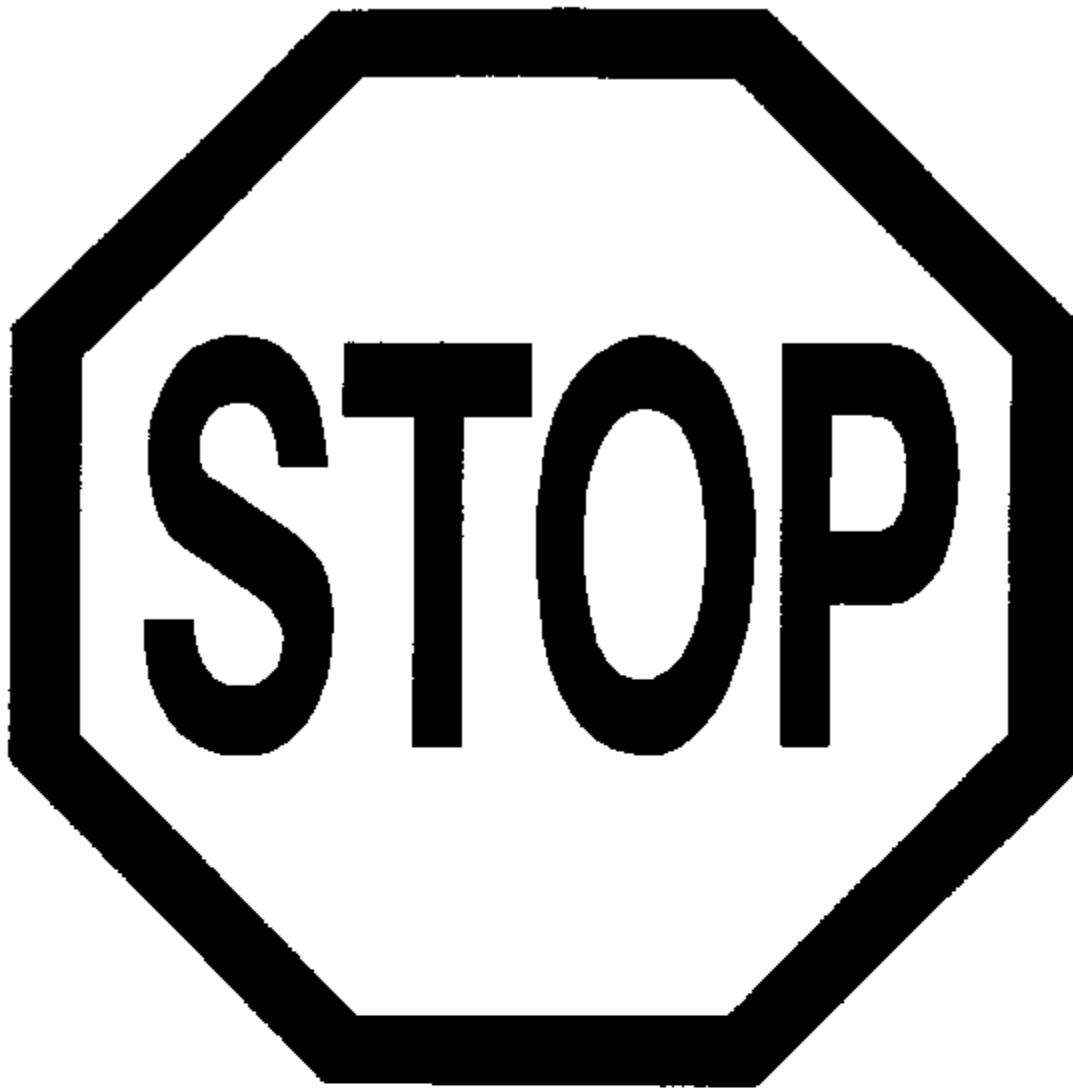
Met vriendelijke groeten,

het Bestuur van de Federatie van Atom ComputerClubs

Laat dat niet gebeuren!

Veel leesplezier met dit nieuwe nummer Atom Nieuws

Namens de redactie, Roland Leurs



BELANGRIJKE MEDEDELING

VOOR LEDEN DIE MET I/O AAN DE SLAG GAAN !

In het handboek van de Atom-in-PC staat in de componentenopstelling een ernstige typefout! De twee adresbuffers zijn aangeduid als 74HCT241. Dit is niet correct. Vervang deze buffers door het type

74HCT541

Onder normale omstandigheden merkt u niets van deze fout, echter de helft van alle adreslijnen zijn "zwevend" op de I/O connector aanwezig, met als gevolg dat er nooit een geldig adres op de externe adresbus staat. Het betreft hier alleen een typefout; de print is ontworpen voor deze 74HCT541. Er hoeft verder niets gewijzigd te worden!

Dan is er nog een fout in de aansturing van de databus buffer. Zoals bekend kan met pen 1 van de 74HCT245 de richting van het datatransport aangegeven worden. Op de print is deze pen aangesloten op pen 14 van GALATPC3. Deze pen is echter niet geprogrammeerd en is altijd hoog. Datatransport loopt dus altijd van Atom naar buiten. Als u gebruik gaat maken van I/O dient deze gal opnieuw geprogrammeerd te worden. Mocht u in het bezit willen komen van deze nieuwe gal, neem dan contact op met Bruno Tossaint of Roland Leurs.

I/O aan de Atom-in-PC

door Roland Leurs
met medewerking van

Bruno Tossaint, Emile Hounjet, Willy Truyen en Henk Bastings.

Inleiding

De I/O kaart voor onze Atom-in-PC begint zijn uiteindelijke vorm aan te nemen. In dit artikel kunt u lezen hoe de werking van het nieuwe I/O systeem is, welke mogelijkheden er zijn en hoe de kaart op het systeem aangesloten moet worden.

Het streven is om een geheel Atom-compatibel I/O systeem op te zetten. Daarbij dan nog het liefst de mogelijkheid tot echt onbeperkt uitbreiden. Tijdens enkele regio avonden in het begin van dit jaar hebben we in de regio Limburg het hier beschreven systeem bedacht.

Bij het bedenken van het nieuwe systeem waren enkele punten heel belangrijk:

- compatibel met de standaard Atom
- beter en onbeperkt uitbreidbaar
- eenvoudig te implementeren op de Atom-in-PC kaart

Deze drie punten spreken natuurlijk voor zich. Compatibiliteit is belangrijk zodat alle bestaande hardware (uitbreidingen) aan de nieuwe Atom gekoppeld kunnen worden. Hierbij denken we aan eerste instantie aan schakelingen die door de VIA bestuurd worden en aan de 8271 disk controller zodat we standaard Atom DOS diskettes aan kunnen.

De (onbeperkte) uitbreidingen betreffen niet alleen I/O toevoegingen maar ook geheugen uitbreiding. Henk Bastings loopt nog steeds met ideeën rond om het gehele Atom geheugen uit te breiden naar bijvoorbeeld 1 MB. Een andere vorm van uitbreiden is de ruisvrije videokaart van Henk. Deze kaart heeft een 6847 aan boord zodat op een aparte monitor de Atom VDU beschikbaar is (leuk voor multi-tasking systemen).

De kunst is om dit alles aan de bestaande kaart toe te voegen, want we willen niet weer terugvallen op de standaard Atom werkwijze: kras die baan door en soldeer een IC op een ander etc. Nee, op de nieuwe kaart moeten zo weinig mogelijk aanpassingen plaats vinden. Hoe we dat gaan doen leest u verderop in dit artikel.

De werking

De nieuwe I/O verschilt op één punt heel veel van de standaard Atom I/O. In het nieuwe systeem is een controle lijn toegevoegd aan de I/O connector. Met behulp van deze lijn kunnen we het hele interne geheugen en I/O van de Atom-in-PC uitschakelen.

Om het op niveau te zeggen:

Als we die lijn 'I/O enable' aan een laag niveau hangen dan is de hele kaart niet meer dan een gebufferde processorkaart.

Hiertoe is GAL ATPC1, de gal die de Atom adresdecodering verzorgt, gewijzigd. Deze gal had al een enable lijn voor het video-geheugen. Door middel van een jumper kon het videogeheugen op de kaart aan of uitgeschakeld worden. Deze functie is nu vervallen. Daarvoor is nu het I/O Enable in de plaats gekomen. Op het moment dat deze I/O Enable laag wordt ($< 0,8$ V) blokkeert de GAL alle uitgangen (lees ChipSelect lijnen). Verder hangt de databus buffer (een 74HCT245) ook aan deze I/O enable lijn, zodat deze opengezet wordt voor datatransport van/naar buiten.

Zodra een uitbreiding geadresseerd wordt zorgt de decoderings logica ervoor dat uiteraard de uitbreiding geselecteerd wordt (CS ingang wordt laag) en tegelijkertijd het I/O enable signaal laag wordt. Dit I/O enable signaal dient een Open Collector uitgang te zijn, aangezien meerdere uitbreidingen het signaal moeten kunnen bedienen. Vergelijk dit met de IRQ lijn waarop ook meerdere IC's aangesloten kunnen worden.

Twee voorbeelden:

- Op de Atom-in-PC kaart zit een 6522 VIA (adresgebied #B800 t/m #B87F. Hiervan worden slechts 16 bytes daadwerkelijk gebruikt (bijv. #B800-#B80F of #B810-#B81F).
Op de I/O kaart is ook een VIA geplaatst in het gebied #B800-#B80F. Het lijkt op dubbele adressering, maar dat is het niet want op het moment dat de externe VIA op #B800 geadresseerd wordt zorgt de I/O Enable er voor dat de VIA op de Atomkaart niet geselecteerd wordt. Deze interne VIA kunnen we aanspreken op bijvoorbeeld #B810 want dan wordt de externe VIA niet geselecteerd en wordt ook de I/O Enable niet actief.
- Een soortgelijk voorbeeld is Henk's videokaart. Als deze kaart aangesloten wordt op de Atom-in-PC zorgt deze kaart er voor dat bij adressering in het geheugengebied #8000 t/m #9FFF (het videogeheugen) het interne geheugen van de Atom geblokkeerd wordt. Op het moment dat deze videokaart aan het systeem hangt is het interne videogeheugen uitgeschakeld. Als de videokaart losgekoppeld wordt werkt het interne geheugen weer.

Een van de grote voordelen van deze uitvoering is dat u een uitbreiding kunt maken en deze aansluiten op het systeem zonder overige hardware wijzigingen te maken. Dus nooit meer doorkrassen en piggy-packen. Mocht een ontwerp of uitbreiding niet werken of niet bevallen dan kunt u altijd terug naar de oude situatie, eenvoudig door alleen de betreffende print er uit te trekken.

Opbouw van de I/O kaart

De I/O kaart bevat een 40-polige connector waarop de aansluiting met de Atom-in-PC gerealiseerd wordt. In de data, adres en controle lijnen zijn weerstanden opgenomen voor demping van reflecties. Het andere uiteinde van de kaart is een 64 polige AB of AC connector met de penbezetting zoals die van de standaard Atom.

Daartussen zijn nog een 6522 VIA aanwezig op de print en een GAL die voor adressering van die VIA zorgt en een aantal bussignalen aanmaakt:

GAL 16V8			
A15	1	20	VCC
A14	2	19	NC2
A13	3	18	A8
A12	4	17	A9
A10	5	16	B80X
A11	6	15	B4XX
A7	7	14	I/O ENABLE
A6	8	13	BXXX
A5	9	12	NC1
GND	10	11	A4

De uitgangen zijn:

B80X	chip select voor de 6522 op de I/O print
B4XX	enable lijn voor PL8 connector (deze PL8 is niet op alle uitvoeringen aanwezig)
BXXX	enable lijn voor Bxxx blok (vervangt BLK0 op de 64 polige connector)
I/O ENABLE	besturing van I/O Enable lijn voor deze kaart. Zie het volgende hoofdstuk voor omschrijving van deze I/O Enable.

Penbezetting en benaming onder voorbehoud !

Wat de voeding betreft zijn er twee mogelijkheden. Ofwel de voeding wordt vanuit de PC gehaald of er komt een externe voeding bij voor de uitbreidingen. Beide methoden hebben voor en nadelen. Een groot nadeel bij gebruik van de PC voeding kan zijn dat de spanning door een kortsluiting wegvalt op een kritisch moment waarbij de harddisk bezig is met schrijfacties. In zo'n geval kan er gegevensverlies optreden. Echter de keuze en verantwoordelijkheid blijft bij u!

De aansluiting voor een luidspreker op #B002 zoals beschreven in AN13-2 is alsnog vanwege plaatsgebrek komen te vervallen.

Compatibiliteit

Het nieuwe I/O systeem is geheel compatibel met de standaard Atom. De I/O print is voorzien van een 6522 VIA de aangesproken wordt in het adresgebied #B800 t/m #B80F. Zoals reeds vermeld is ook de 64 polige connector compatibel met de standaard Atom connector. Een kleine uitzondering hierop is het BLK0 signaal. Dit is vervangen door het BXXX signaal van bovengenoemde GAL. Velen hebben deze aanpassing ook al gemaakt in de oude Atom. Als iemand hier problemen mee heeft dan kan dat altijd gewijzigd worden door de GAL opnieuw te programmeren.

Aangezien een groot aantal uitbreidingskaarten voor de Atom al geplaatst zijn in het gebied #BC00-#BFFF wordt het I/O Enable signaal al actief gemaakt door de GAL. Dat betekent dat bestaande printen in dat adresbereik niet gewijzigd hoeven te worden. Hetzelfde geldt voor uitbreidingen in het gebied #B000 t/m #B7FF.

In "formule-vorm":

$$/IO-Enable = /(\#B000 \text{ t/m } \#B80F) + /(\#BC00 \text{ t/m } \#BFFF)$$

Het aansluiten van de I/O kaart

Het meest belangrijke punt, het aansluiten van de I/O kaart en het wijzigen van de Atom-in-PC. Zoals beloofd gaan we niet krassen, maar de soldeerbout moet even aan (alleen nodig als U de I/O-kaart plaatst). Zie bijlage 4.

Verwijder de weerstand bij de jumper (JP2) recht onder de 40-polige I/O connector, als deze geplaatst is . Plaats de jumper op de contacten 1-2. Verwijder de jumper JP1 bij de adresseringsgal (GALATPC1). Leg een draad van de bufferaansturingsjumper JP2(contacten 1-2) naar het middencontact van de verwijderde jumper van JP1.

Verwijder GALATPC1 en vervang deze door een nieuwe versie. Plaats de Atom daarna weer in de PC en deze moet dan weer als vanouds werken. Om te testen of de I/O Enable werkt kunt u pen 37 van de 40 polige I/O connector eens met een draadje aan de 0V hangen. De Atom-in-PC moet zich dan ophangen.

Voordat de I/O kaart aangesloten wordt moet ook het operating systeem vervangen worden. Het operating systeem gaat er immers van uit dat de PC-communicatie VIA op adres #B800 geplaatst is. Op een kale Atom-in-PC is dat wel zo, maar met de I/O kaart komt de "ATOM-I/O- VIA" op dat adres. Daarom gaat de PC-communicatie VIA "verhuizen" naar #B810 t/m #B81F. Deze verhuizing is puur softwarematig.

In een volgende versie van het operating systeem 3.09 zal deze wijziging uitgevoerd zijn. U kunt de wijzigingen zelf aanbrengen door de onderstaande adressen aan te passen:

Adres	is	wordt
#ED36	#03	#13
#ED3A	#02	#12
#ED3F	#0C	#1C
#ED42	#01	#11
#EDC1	#0D	#1D
#EDC8	#01	#11
#EDCD	#00	#10
#EDD0	#0D	#1D

Denk eraan dat deze wijziging (ook) in de EPROM wordt uitgevoerd. Anders zal de Atom-in-PC nooit kunnen communiceren met de pc als de I/O print aangesloten is.

Nogmaals : vanaf operating-systeem versie 3.09 is deze wijziging opgenomen; als uw kaart na 1 september 1994 is geleverd, dan heeft u deze versie al in huis, evenals de aangepaste gal GALATPC1.

Huidige status

Het prototype is inmiddels gemaakt en getest. Hierbij zijn op een klein detail na (ergens staat een 2 i.p.v. een 5 op een bepaald IC, zie ook elders in dit nummer) geen problemen naar voren gekomen. Er vinden nu alleen nog wat kleine wijzigingen plaats in de opstelling, om de print wat gemakkelijker op te bouwen (al verwerkt in de bijlage) Daarmee is dit ontwerp dan af en kan het op de ATOM-MARKT verschijnen.

Vanzelfsprekend zijn op de landdag de demo's te zien.

Tot zover het verhaal over de I/O voor de Atom-in-PC. Afgezien van wat kleine software wijzigingen in de toekomst is het project Atom-in-PC hiermee zo goed als afgerond. Het is nu uw beurt om wat te doen met die kaart. We zien uw programma's, uitbreidingen etc binnenkort binnenstromen bij de redactie.

Het schema en printontwerp werd uitgevoerd door Bruno, Emile zorgde voor de dubbelzijdige print, Willy zorgde voor de afwerking, en last but not least Henk voor de gal.

Als er iets niet duidelijk is dan horen we het wel,
Succes met bouwen,
met de vriendelijke groeten van Roland en de anderen.

(voor bijlagen en opmerkingen zie de volgende pagina).

- * Bijlage 1 : De veranderingen in de ATOM-in PC.
- * Bijlage 1a: De veranderingen in het ATOM-in PC schema
- * Bijlage 2 : Het schema van de I/O-kaart.
- * Bijlage 2a/b:Het print-ontwerp voor de I/O kaart

- n.b. 1. Het printontwerp is uitgevoerd als dubbelzijdige print. Voor diegenen die zoiets zelf kunnen vervaardigen is uiteraard de originele file of evt. een printfilm beschikbaar. Vooralsnog denken wij dat het aantal gegadigden beperkt zal blijven, weshalve er geen officiële print wordt aangemaakt. Het aantal doorverbindingen is n.l. beperkt. Alleen het solderen van de A/B of A/C-connector is een "klein kunstje", de deskundigen zijn gaarne bereid te vertellen hoe dat moet. Als echter zou blijken dat er een groter aantal gegadigden zijn, zo in de buurt van de 20, wordt het wellicht mogelijk ook hier professionele versie van te maken.
2. Degenen die hun eprom's en gal's niet zelf willen of kunnen aanpassen, danwel de GAL ATPC4, voor de I/O kaart willen bestellen, kunnen contact opnemen met de redactie.

REGIO-MEDEDELINGEN.

1. REGIO BRABANT-OOST

Bijeenkomsten op het bekende adres :
Adolf van Cortenbachstraat 92, Eindhoven, tel. 040-123231.
Aanvang 13.30.u

2. REGIO LIMBURG-BELGIE

Clubavonden in "Oos Kaar", Geldersestraat 43, tel 046-321378.
op de 1e vrijdag van de maand.

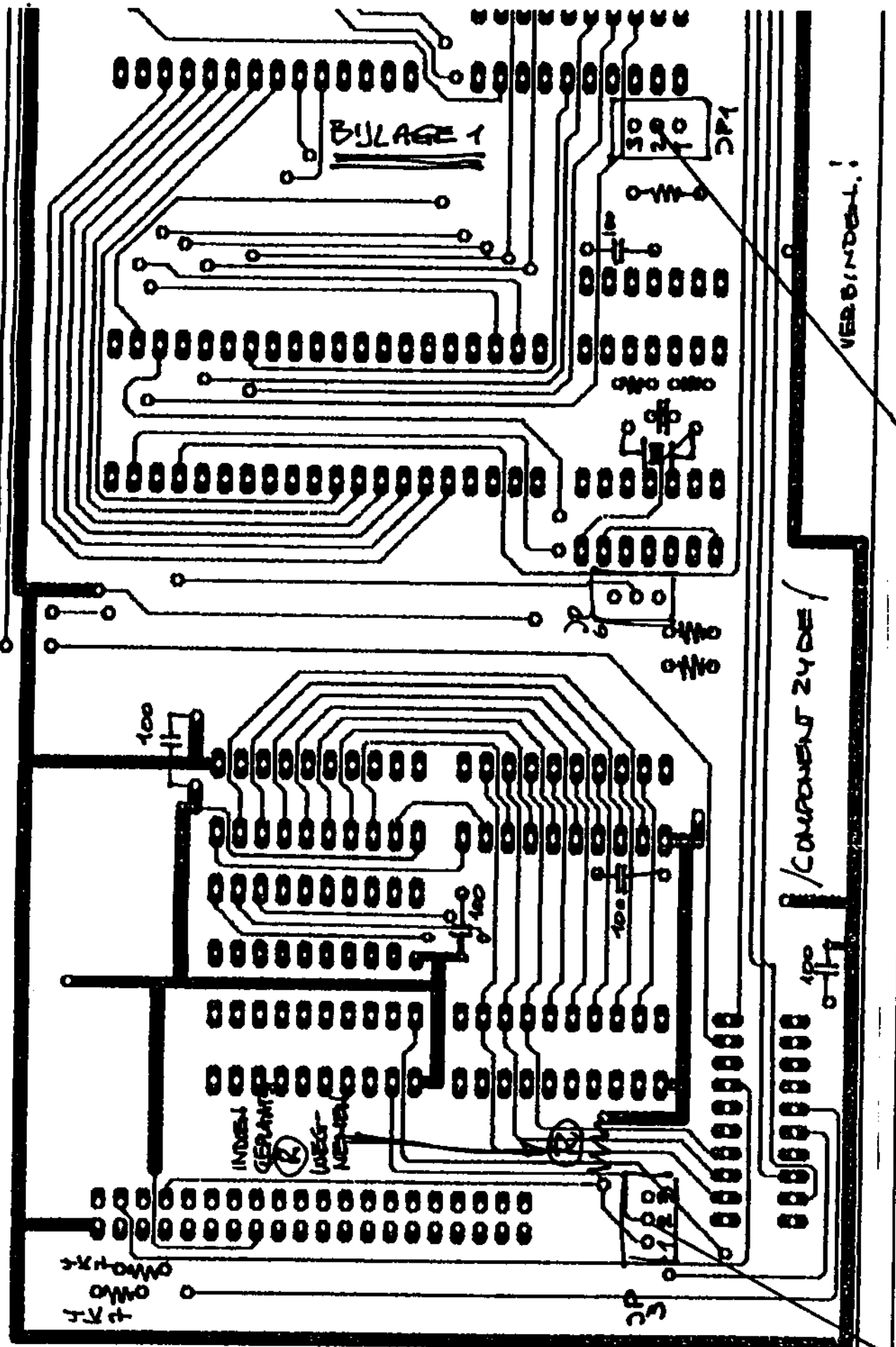
3. REGIO DEN HAAG

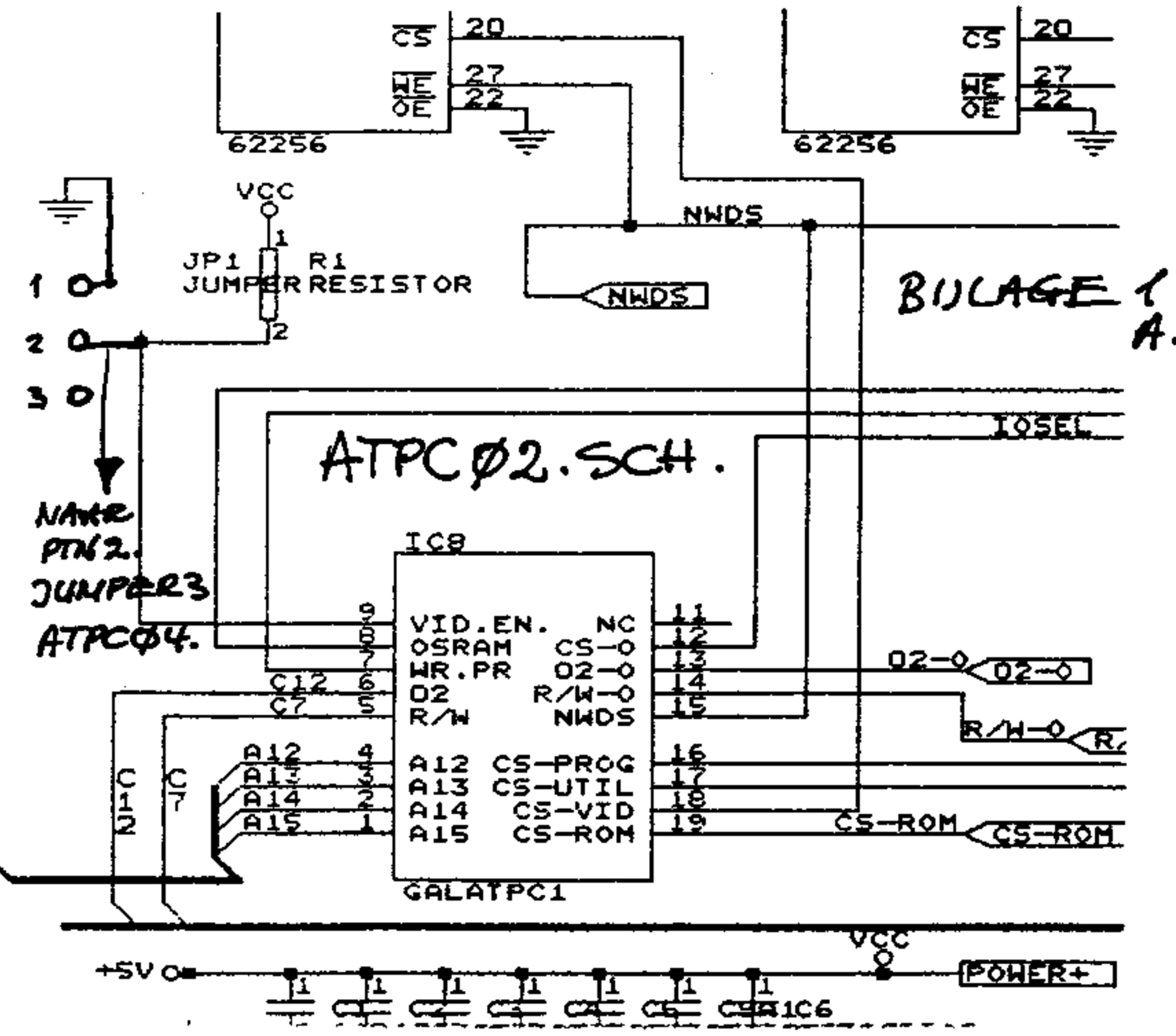
Data voor de regio-avonden in 1994 :
28 Oct., 09 Dec.
Alles op het nieuwe adres : Theo Waayer
Hendrik v. Boeijenlaan 66, 2273 DC Voorburg, t. 070-3862504

4. REGIO ARNHEM e.o.

Geplande bijeenkomsten Acorn Atom Club Regio Arnhem:
Ten Huize van Henri Derksen, Bolwerk 25, 6811 JE ARNHEM
20 Oct., 17 Nov., 15 Dec., dit is de derde woensdag vd maand
Indien er een landelijke dag in de maand is, vervalt de
regionale bijeenkomst.

Telefoon: 085-455485, UniCorn BBS: 085-425506 xxxx/xxxx BPS
8N1

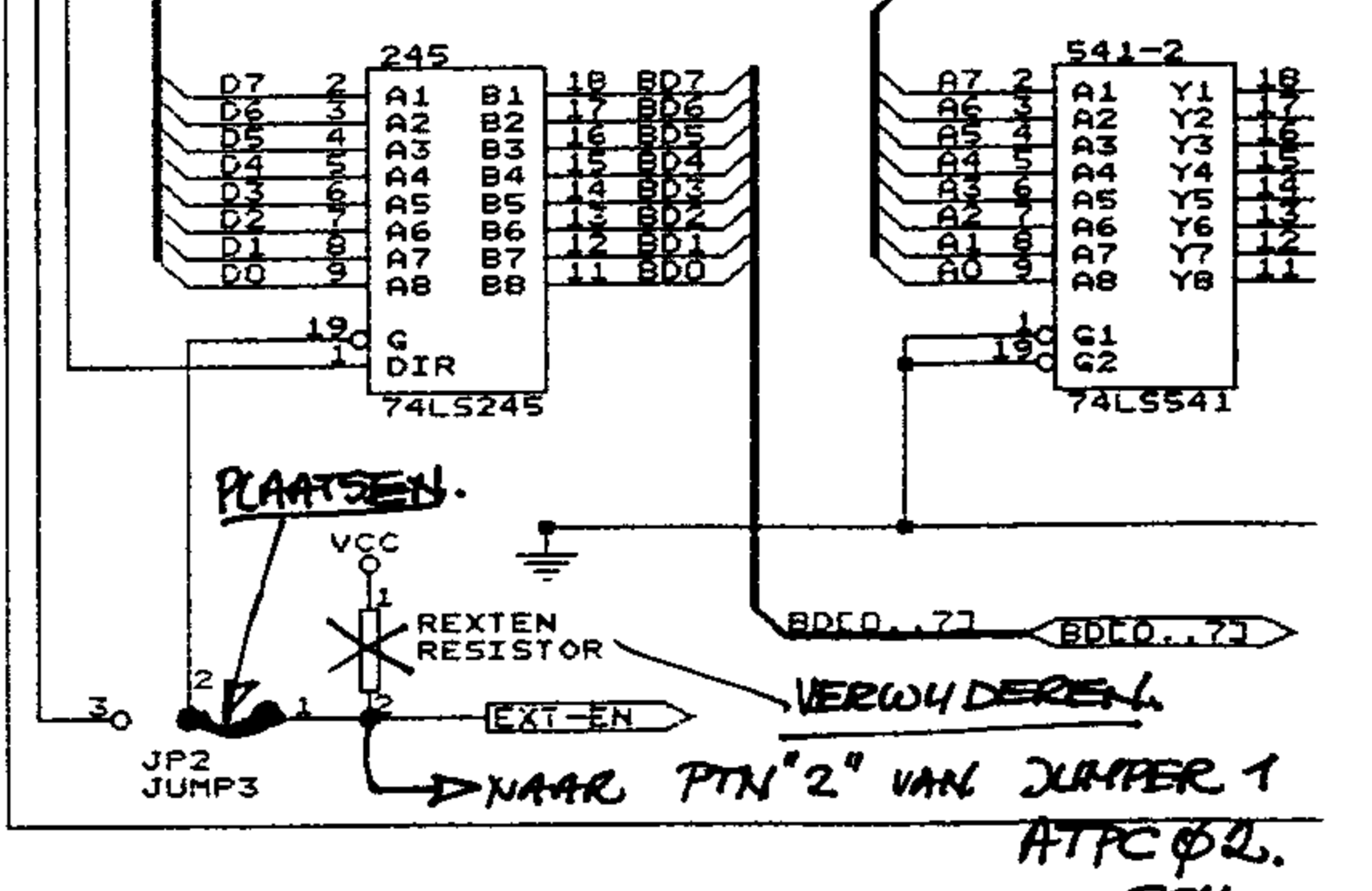


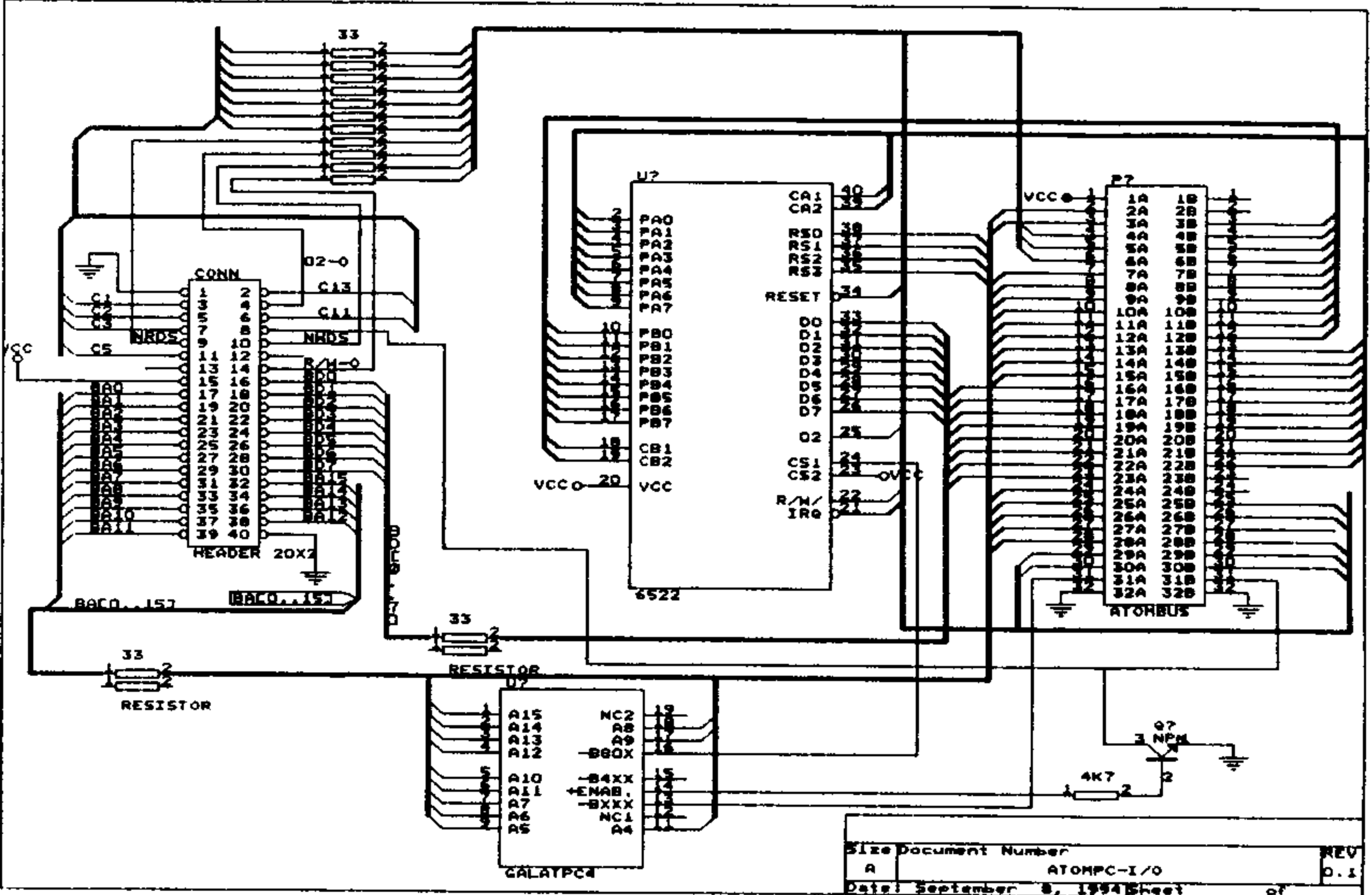


BILAGE 1 A.

NAME PTN 2. JUMPER 3 ATPC Ø4.

ATPC Ø4. SCH.

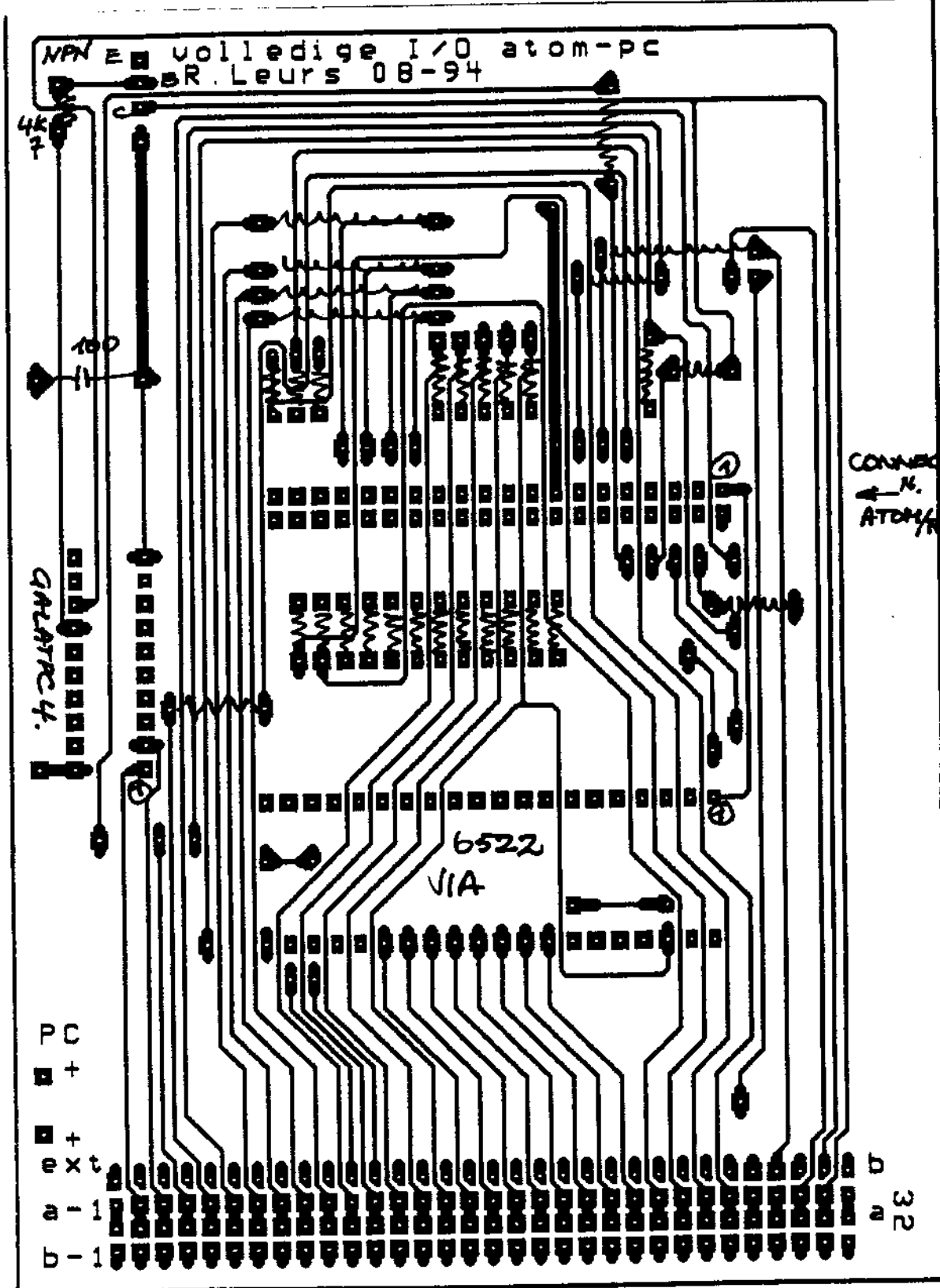




BULAGE 2

Size Document Number	REV
A	D.1
Date: September 8, 1984	Sheet 07

BULAGE 2a



NPN E volledge I/O atom-pc
 R. Leurs 08-94

4K7
 7

100

ATOMPC4.

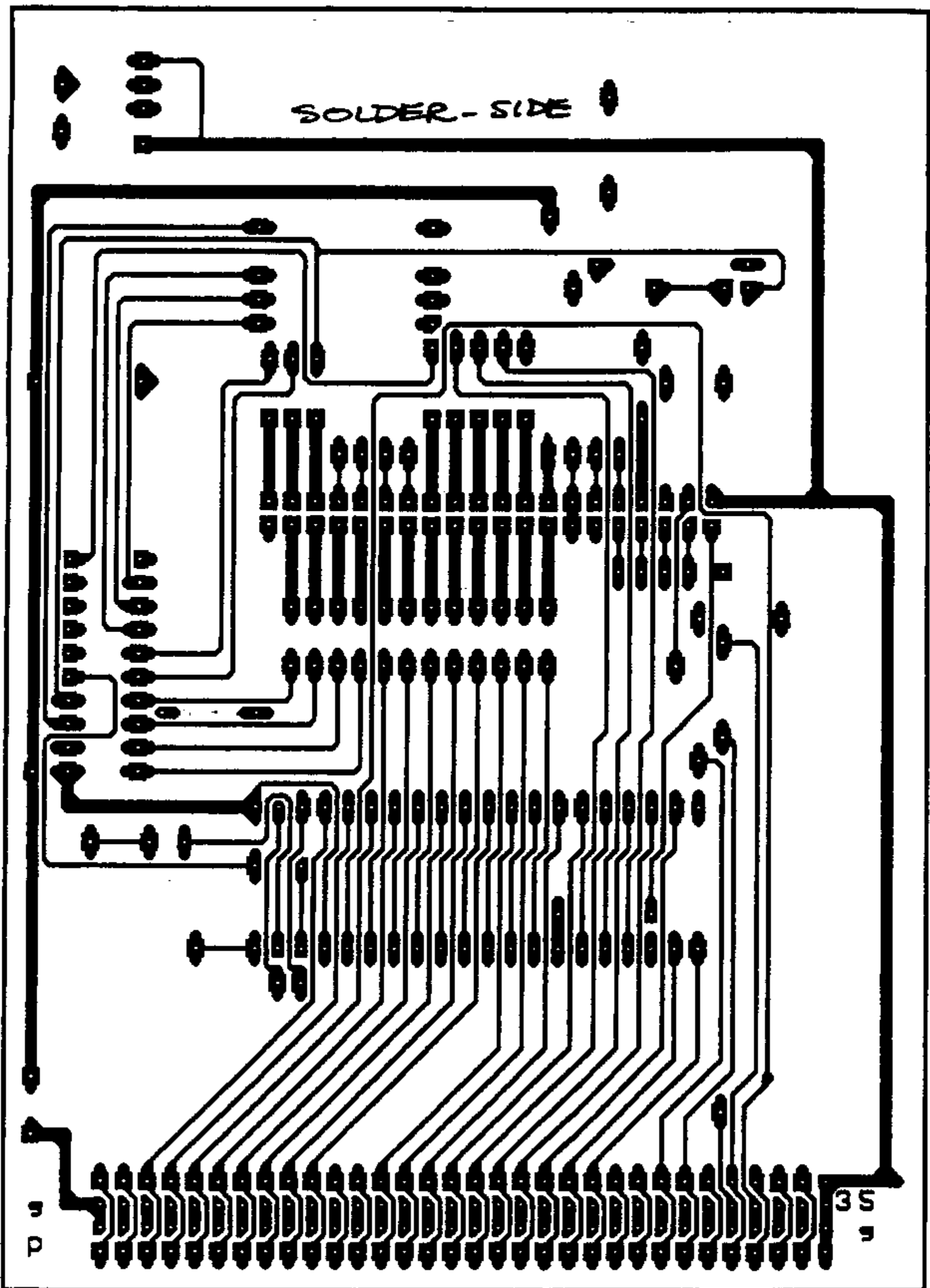
6522
 VIA

CONNECT.
 ← N.
 ATOMPC

PC
 +
 +
 e X 2
 e 1 1
 U 1 1

32
 D N

BULAGE 2b




```

=====
=           A T O M T A L K           =
=====

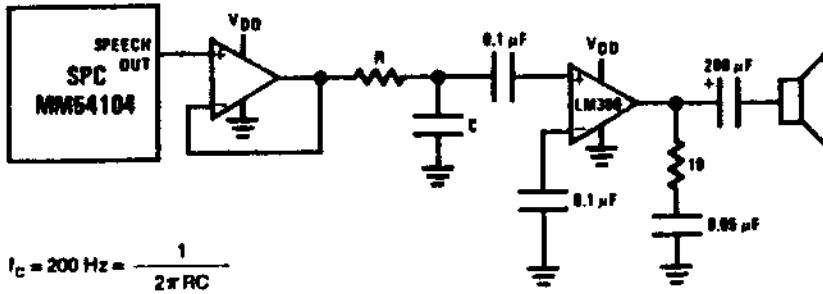
```

Vroeger toen de firma Tandy nog bestond heb ik, naar aanleiding van een artikel in Atom Nieuws 9-2 blz. 32 door Leo Gijssel, de daar aangeprezen chip CTS256A-AL2 gekocht. Ik vond dat toen een interessante chip maar door tijdgebrek (waar heb ik dat meer gehoord?) is het er nu pas van gekomen. Leo had toen beloofd het schema nog eens te tekenen, maar dat heb ik niet meer van hem gezien. Onlangs heb ik ATOMTALK gebouwd en het werkt zoals in het artikel beschreven is. Ik wil dan ook de werking niet meer opnieuw uitleggen, daar is het artikel van Leo zeker voldoende voor. Wel wil ik nog een paar dingen van het bijgaande schema verduidelijken. De ASCII code komt via printeraansluiting J1 en buffer U6 in de inputbuffer van U1 en als die vol is, in de extensie RAM U3. Als de RAM vol is geeft U1 een busy signaal via U5E op de busy pin van J1 en led D4 gaat aan. U1 zet de ASCII code om in foneem code en wel zodanig dat normale Engelse tekst voor 85% goed uitgesproken wordt. Dit volgens de data sheet van Tandy. Deze foneem code wordt als data doorgegeven naar de outputbuffer van U1 en via buffer U2 naar de eigenlijke foneemgenerator U7. Dit is de alom bekende SP0256A-AL2 chip. Dit IC zorgt dan voor het audio signaal. Om de verstaanbaarheid optimaal te doen zijn is uitgebreide filtering toegepast. Een en ander volgens een data sheet van een speech synthesizer van National Semiconductor. Zie bijgaande schema's en audio karakteristiek. Als output IC is de bekende LM386 toegepast. Speciale aandacht voor C25. In applicatieschema's komt die nooit voor, maar is wel effectief. Deze condensator voorkomt radio ontvangst als de bedrading wat lang is. Moet kort tussen de pinnen van het IC gemonteerd worden. Probeer maar eens bij de LM386.

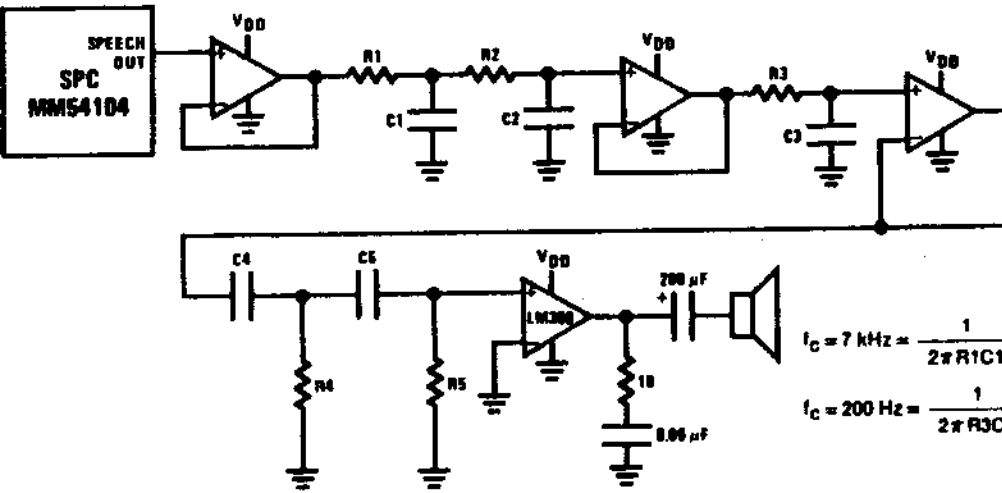
U1, de CTS256A-AL2 verzorgt de gehele besturing van de spraakchip, ook de besturing van de buffers en de RAM buffer. Verder blijkt er ook nog een clockoutput te zijn, welke een clockfrequentie gedeeld door 4 geeft. In het TANDY schema wordt die niet gebruikt. Door voor de CTS256A-AL2 een kristal van 12.48 MHz te nemen is de clockfrequentie van de SP0256A-AL2 precies 3.12 MHz en dus de "pitch" juist. Dus geen kinderstemmetje zoals met 3.57 MHz. Bovendien een kristal minder. S1 is de mode schakelaar. 1, 2 en 3 staan op 0 voor parallel input mode, 4 doet alleen mee in serie mode, 5 is 1 voor externe RAM buffer en 6 is 0 als op een carriage-return gewacht moet worden om te spreken. Als 6 is 1 spreekt ATOMTALK na elke "delimiter". Bij het inschakelen van de voeding of bij een reset meldt ATOMTALK zich met "OK". De aansturing gebeurt gewoon met P.\$2 en P.\$3. Alles wat na P.\$2 wordt getypt probeert ATOMTALK uit te spreken. P.\$3 maakt hem rustig.

Op de landdag van april 1994 heeft U ATOMTALK kunnen horen habbelen!

Het TANDY applicatieschema is in het archief aanwezig.



(a) Minimum Low-Pass Filter/Amplifier



(b) Maximum Filter Response Configuration

FIGURE 4. SPC Filter and Amplifier

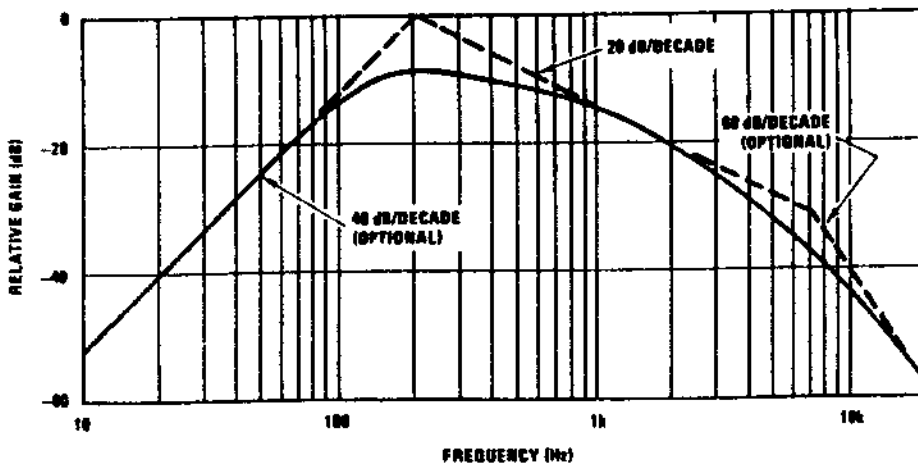


FIGURE 5. Recommended Frequency Response of Entire Audio System for MM54104 SPC



TECHNICAL DATA

AN EXCLUSIVE RADIO SHACK SERVICE TO THE EXPERIMENTER

CTS256A-AL2 CODE-TO-SPEECH PROCESSING CHIP

Features:

- Unlimited vocabulary
- Utilizes letter-to-sound rules
- Serial or parallel interface
- Outputs directly to the SP0256A-AL2 Speech Processor
- User can add to existing letter-to-sound rules

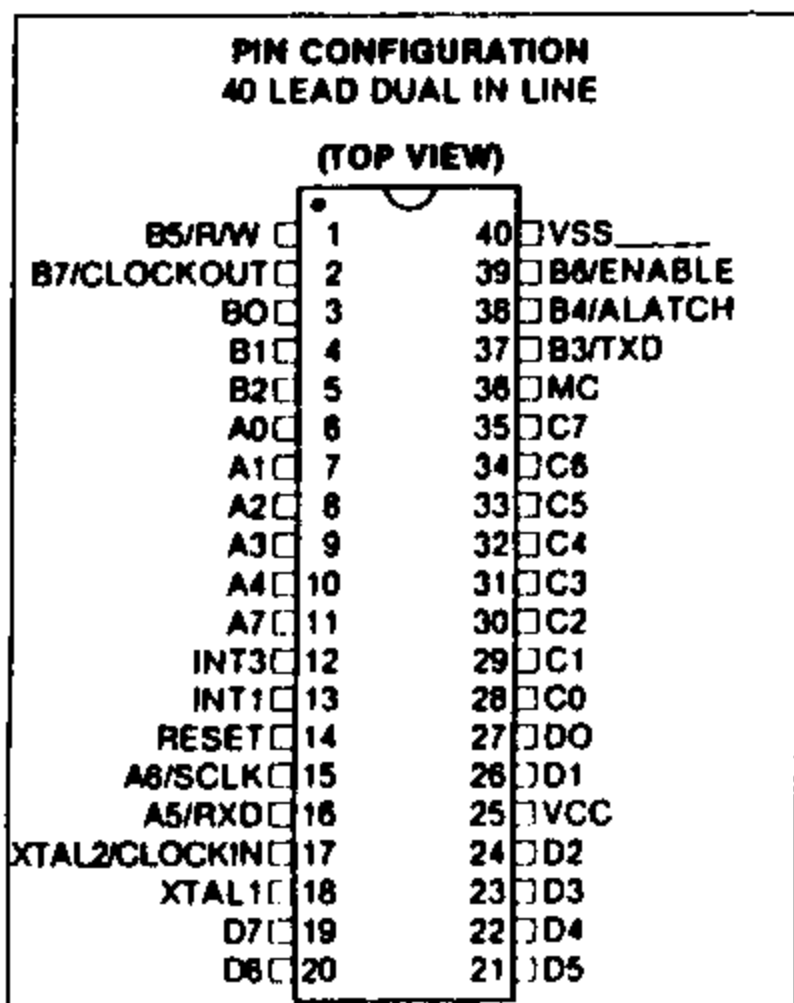
General Description

The CTS256A-AL2 is an 8-bit microprocessor programmed with letter-to-sound rules. This built-in algorithm allows for the automatic real-time translation of English ASCII characters into alphanumeric addresses compatible with the SP0256A-AL2 (276-1784) Narrator® Speech Processor.

The CTS256A-AL2 is functionally equivalent and pin-for-pin compatible with the PIC7041. As such, it retains its ability as an 8-bit microcomputer for execution of user-defined programs resident externally. Its uniqueness is the internal code-to-speech algorithm which converts English text (in the form of standard ASCII characters) into alphanumeric addresses, using letter-to-sound rules.

The chip can be used with virtually any microcomputer, since it utilizes either standard serial (TTL Level) or parallel input, with full hardware (DTR-DSR) and software (XON-XOFF) handshaking.

Upon power-up or use of the hardware reset, the system initializes itself and then speaks 'O.K.' to demonstrate that it is ready for input.



ELECTRICAL CHARACTERISTICS

Maximum Ratings*

All pins with respect to Vss -0.3 to 8.0V
Storage Temperature -25°C to 125°C

Standard Conditions

Clock—Crystal Frequency 10.00 MHz
Operating Temperature (T) 0°C to 70°C

DC CHARACTERISTICS/CTS256A-AL2

Exceeding these ratings could cause permanent damage to the device. This is a stress rating only and functional operation of this device at these conditions is not implied. Operating ranges are specified in Standard Conditions. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Data labeled "typical" is presented for design guidance only and is not guaranteed.

**TABLE 1
SPECIAL PIN FUNCTIONS**

INPUT MODE SELECTION				
PIN	6	7	8	MODE
	0	0	0	PARALLEL INPUT MODE
	0	0	1	SERIAL MODE - 50 BAUD
	0	1	0	SERIAL MODE - 110 BAUD
	0	1	1	SERIAL MODE - 300 BAUD
	1	0	0	SERIAL MODE - 1200 BAUD
	1	0	1	SERIAL MODE - 2400 BAUD
	1	1	0	SERIAL MODE - 4800 BAUD
	1	1	1	SERIAL MODE - 9600 BAUD
PIN	9			
	0	DEFAULT SERIAL PARAMETERS (7 BITS, 2 STOP BITS, NO PARITY)		
	1	SELECTABLE SERIAL PARAMETERS (SEE TABLE 2)		
PIN	10			
	0	INTERNAL RAM BUFFERS (20 BYTE INPUT/26 BYTE OUTPUT)		
	1	EXTERNAL RAM BUFFERS (1972 BYTE INPUT/256 BYTE OUTPUT)		
PIN	11			
	0	CARRIAGE-RETURN ONLY DELIMETER		
	1	ANY DELIMETER		
PIN	3	(OUTPUT ONLY)		
	0	INPUT BUFFER IS "BUSY"		
	1	INPUT BUFFER IS "NOT BUSY"		

**TABLE 2
U10 (74LS373) UART CONFIGURATION**

PIN	13	14	
	0	1	6 BITS/CHARACTER
	1	0	7 BITS/CHARACTER
	1	1	8 BITS/CHARACTER
PIN	8		
	1	PARITY ENABLED	
	0	PARITY DISABLED	
PIN	7		
	1	EVEN PARITY	
	0	ODD PARITY	
PIN	3		
	0	ONE STOP BIT	
	1	TWO STOP BITS	

**** EEN DISPLAY AAN DE ATOM ****

Sinds enige tijd zijn er in de handel LCD-displays te krijgen met een standaard aansluiting en programmering. Deze displays zijn gemakkelijk aan een computersysteem te koppelen en in dit artikel zal ik de hardware en software beschrijven om deze displays aan de Atom te koppelen.

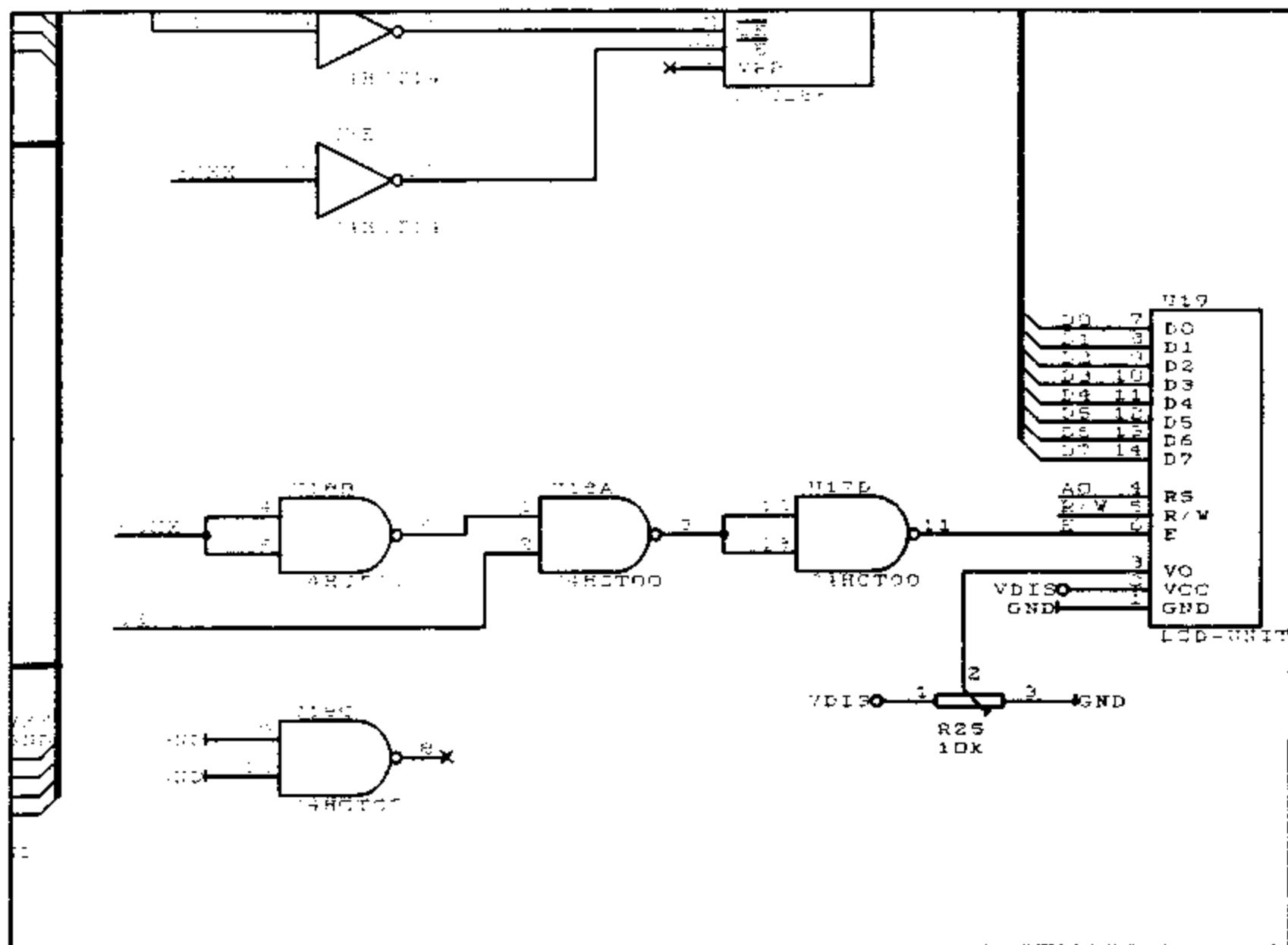
DE HARDWARE

De displays zijn uitgerust met één of twee regels van 16,20 of 40 karakters. Deze karakters zijn opgebouwd uit 7*5 of 10*5 dots en een cursorregel. De displays zijn uitgerust met een controller van b.v. Hitachi, Oki of JRC en een aantal drivers voor het display (aantal afhankelijk van het aantal karakters). De displays worden o.a. gefabriceerd door Hitachi en Philips (Heerlen). De controllers en dus ook de displays van de verschillende fabrikanten worden op dezelfde manier geprogrammeerd.

De displays hebben 14 aansluitpennen, uitgevoerd als een 14 pins single-in-line of een 2*7 pins flatcable-connector. (Sommige displays hebben 16 aansluitingen, waarbij pen 15 en 16 voor de achtergrondverlichting worden gebruikt). In de figuur op de volgende bladzijde is een stuk van het schema van de voorversterker getekend. Rechts is de LCD-unit getekend en deze heeft de volgende aansluitingen:

- de acht datalijnen
- A0 op RS, het display heeft dus twee adressen in de memory map; op het even adres een commando register en op het oneven adres een dataregister.
- R/W op R/W
- E is het actief hoge (!) enable signaal. Hierbij wordt de adres-selectie met een AND gesynchroniseerd met Q2. In het schema is BCXX actief laag (komt van een 74HCT139). Dit signaal wordt dus eerst geïnverteerd, hierna "ge-AND" met Q2 en dit actief lage signaal wordt weer geïnverteerd. Hierdoor heb ik maar één extra IC nodig, U18 dus.
- VO is de pen voor de contrast regeling, op deze pen wordt d.m.v. een potmeter 0 tot 5 volt aangeboden, waarmee dus het contrast kan worden geregeld. Bij een 40 karakter display voldoet dit minder, het display kan niet "vol open" worden gezet. Als oplossing moet de spanning op VO onder 0 volt worden gebracht, dit staat beschreven in Elektuur 6-93.
- VCC = 5 volt
- GND = 0 volt

Bij sommige displays zijn dus nog twee pennen voor achtergrondverlichting aanwezig. Van de door mij toegepaste Philips displays LTN211 (Grootaerds Maastricht fl 49,-) hebben sommige een achtergrondverlichting met aparte aansluitingen. De achtergrondverlichting kan bestaan uit LED's of uit een platte strip waarop dan 110 volt AC moet worden gezet (soort neon of fluorescentie lamp ?). Hiervoor worden aparte convertortjes gebruikt om van 5V DC 110V AC te maken. De groene LED achtergrondverlichting geeft echter het mooiste resultaat.



Al met al is het hardware gedeelte dus, zoals meestal bij de Atom, zeer eenvoudig. Als enigste probleem is het verkrijgen van een 14 pins single-in-line connector (of een 28 pins dubbele) voor het aansluiten van het display. Als iemand weet waar ik een zo'n 14/28 pins connector kan kopen had ik dat graag vernomen.

Op het display zelf zitten een aantal geheugenplaatsen, men kan namelijk zelf karakters definiëren en deze kunnen dan in C(harakter) G(enerator) RAM worden weggeschreven. Verder is het display voorzien van D(isplay) D(ata) RAM. In deze RAM worden de codes weggeschreven, die het karakter voorstellen dat op het display moet worden weergegeven. Deze codes worden d.m.v. een ROM op het display vertaald in een karakter.

De ROM van een controller wordt door de fabrikant volgezet, zodat er van een controller bijvoorbeeld verschillende versies bestaan, omdat de ROM anders geprogrammeerd is. Gelukkig zijn de normale letters, cijfers en leestekens ASCII gecodeerd. Verder kent een display dan nog bijvoorbeeld pijltjes en chinese of japanse tekens (?).

Bij een éénregelig display heeft het eerste karakter DD RAM adres #00, het tweede adres #01 enz. Bij een tweeregelig display is dit bij de eerste regel hetzelfde, de tweede regel begint echter met adres #40 en het tweede karakter op de tweede regel heeft dan adres #41 enz. Bij mijn LTN211 is dat dus als volgt:

```

karakter:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
#adres: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

```

Lijn 2

```

karakter:  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
#adres: 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F

```

Dan volgt nu:

DE SOFTWARE

Het display moet geprogrammeerd worden door het commando's te geven. Deze commando's worden vaak in een tabel weergegeven, bijvoorbeeld in Elektuur of zoals de tabel hieronder uit de datasheet van de LTN211. Hierin zie je dat alle commando's naar het even adres worden geschreven en de data naar het oneven adres (RS=A0).

Aangezien de LTN211 een display is met een tweeregelig display met 7*5 dots per karakter zijn deze instellingen vaste waarden geworden in de tabel. In de uitwerking van de commando's wordt echter de gehele instructie FUNCTION SET beschreven.

Na de uitwerking van de instructies volgt nog de initialisatie van het display en een programmeervoorbeeld om het display op een eenvoudige manier in te passen in een bestaande applicatie.

INSTRUCTION	ADDRESSES/DATA									
	RS	Rw	D7	D6	D5	D4	D3	D2	D1	D0
display clear	0	0	0	0	0	0	0	0	0	1
cursor home	0	0	0	0	0	0	0	0	1	*
entry mode set	0	0	0	0	0	0	0	1	ID	S
display on/off control	0	0	0	0	0	0	1	D	C	B
cursor display shift	0	0	0	0	0	1	SC	RL	*	*
function set	0	0	0	0	1	DL	1	0	*	*
CG RAM address set	0	0	0	1	ACG					
DD RAM address set	0	0	1	ADD						
Busy flag/adress read	0	1	BF	AC						
CG RAM/DD RAM data write	1	0	write data							
CG RAM/DD RAM data read	1	1	read data							

ID = 1 ; increment	ID = 0 ; decrement
S = 1 ; display shift	S = 0 ; display freeze
D = 1 ; display on	D = 0 ; display off
C = 1 ; cursor on	C = 0 ; cursor off
B = 1 ; character blinks	B = 0 ; char does not blink
SC = 1 ; display shift	SC = 0 ; cursor move
RL = 1 ; right shift	RL = 0 ; left shift
DL = 1 ; 8 bits	DL = 0 ; 4 bits
BF = 1 ; internal operation	BF = 0 ; end internal operation

Hieronder worden de instructies apart besproken om de duidelijkheid te krijgen, die na het bekijken van de tabel wellicht niet is gekregen. Van elke instructie is de hexadecimale waarde gegeven. Door de ACCU nu met dit getal te laden en dit daarna weg te schrijven naar het commando register kan een instructie worden gegeven.

**** display clear = #01**

- spatie code (in ASCII \$20) in ieder DD ram adres.
- AdresCounter = DD ram adres 0, dus inhoud AC : #00.
- entry mode = increment (zie entry mode)
- Een eventuele cursor of het knipperen van het karakter blijft aan staan. Dit gebeurt dus nu op adres \$00.

**** cursor home = #02**

- AdresCounter = DD ram adres 0, dus inhoud AC : #00
- Als er geshift is, wordt dit shiften ongedaan gemaakt (zie cursor/display shift).
- Een eventuele cursor of het knipperen van het karakter blijft aan staan. Dit gebeurt dus nu op adres \$00.

**** entry mode set = #04 / #05 / #06 / #07**

*** decrement, no shift = #04**

- Als er een waarde in het dataregister wordt geschreven, dan wordt het karakter, wat bij deze waarde hoort op het display gezet en wel op de plaats waar de adrescounter naar wijst! In deze entry mode wordt de adrescounter dan verlaagd met 1. Hierdoor wordt bij een volgende keer wegschrijven van data het karakter links van het vorige veranderd. Je hoeft dus niet per karakter de adrescounter zelf aan te passen, je kan dus achter elkaar de verschillende letters van een bepaalde boodschap wegschrijven.

*** decrement, shift = #05**

- Hierbij schuift de controller de voorgaande letters een plaats op, het lijkt dus of de letters vanuit één bepaalde plaats op het display komen. Bij decrement zullen de letters dus naar links wegschuiven.

*** increment, no shift = #06**

- Als #04, echter wordt de AC verhoogd met 1, waardoor het volgende karakter rechts van het vorige wordt weggeschreven.

*** increment, shift = #07**

- Als #05, echter de letters schuiven naar rechts.

** display on/off control

Hiermee kan je het display, de cursor of het knipperen van een karakter onafhankelijk van elkaar aan en uit zetten. De cursor en het knipperen zijn altijd bij het karakter waar de adrescounter naar wijst! Display uit betekent alleen dat er niets wordt weergegeven, niet dat het display echt "uit" wordt gezet, voor dat laatste moet je de spanning eraf halen.

- * display uit, cursor uit, char knippert niet = #08
- * display uit, cursor aan, char knippert niet = #0A
- * display aan, cursor uit, char knippert niet = #0C
- * display aan, cursor aan, char knippert niet = #0E
- * display uit, cursor uit, char knippert = #09
- * display uit, cursor aan, char knippert = #0B
- * display aan, cursor uit, char knippert = #0D
- * display aan, cursor aan, char knippert = #0F

** cursor display shift

Hiermee kan de cursor of het display worden verschoven. Dit gebeurt dus steeds één keer als de instructie wordt gegeven.

- * cursor shift left = #10

- hierbij wordt de cursor naar links verschoven, de cursor kan van de eerste regel naar de tweede regel worden geschoven en terug. De adrescounter wordt aangepast:
AC = AC - 1

- * cursor shift right = #14

- als boven, de cursor schuift nu naar rechts

- * display shift left, cursor volgt = #18

- hier wordt de hele tekst op het display naar links verschoven. De tekst kan echter maar op één regel blijven, karakter 1 komt terecht op de plaats van 16 en karakter 17 op de plaats van karakter 32. De tekst wordt dus slechts horizontaal verschoven. Hierbij moet worden opgemerkt dat de adrescounter niet verandert. De verandering door het shiften kan ongedaan worden gemaakt met een cursor home.

- * display shift right, cursor volgt = #1D

- als boven, de tekst schuift nu naar rechts.

** function set

- dit is de eerste instructie die het display moet krijgen. Je kan hiermee instellen of de dataoverdracht met 4 of met 8 bits moet gaan, of het display met één of twee regels is opgebouwd en of een karakter is opgebouwd uit 5*10 of 5*7 dots. Hierbij is de instructie opgebouwd uit de volgende bits:

D7	D6	D5	D4	D3	D2	D1	D0
code:	0	0	1	DL	N	F	* * *

DL = 1 ; 8 bits, zoals bij de Atom

DL = 0 ; 4 bits

N = 0 en F = 0 ; 1 regel met 5*7 dots per karakter

N = 1 en F = 1 ; 1 regel met 5*10 dots per karakter

N = 1 en F = 0 ; 2 regels met 5*7 dots per karakter

* = don't care

* Bij de LTN211 is dat dan #38; display werkt met 8 bits, 2 lijnen en 5x7 dots.

** CG RAM adres set = #40 + adres

- met deze instructie kan je het Karakter Generator adres opgeven.
- bij #40 moet een vijf-bits adres worden opgeteld.
- dit adres wordt in de adrescounter gezet
- na deze instructie wordt de data in CG ram gezet EN NIET IN DD RAM!

** DD RAM adres set = #80 + adres

- met deze instructie kan je het Display Data adres opgeven, de plaats dus waar het karakter wordt neergezet na het schrijven van data in het dataregister.
- bij #80 moet een zes-bits adres worden opgeteld.
- dit adres wordt in de adrescounter gezet
- na deze instructie wordt de data in DD ram gezet EN NIET IN CG RAM!

** busy flag/adres read

Nu wordt het commando register gelezen, hierbij geeft dan bit 7 de status van het display aan. Als bit 7 hoog is, dan is het display bezig en kan het geen data of commando's ontvangen, als bit 7 laag is kan het dat weer wel. Voor het versturen van data of een commando naar het display moet je dus eerst deze busy flag controleren! De resterende bits geven het adres van CG of DD RAM, afhankelijk of de vorige instructie van toepassing was op CG of DD RAM.

** CG RAM/DD RAM data write

- wegschrijven van de data=karaktercode op een bepaald adres
- schrijven naar dataregister, selectie CG/DD RAM volgt uit vorige instructie
- het adres wordt automatisch verlaagd/verhoogd volgens entry mode en het display shift naar gelang entry mode

** CG RAM/DD RAM data read

- lezen uit dataregister, selectie CG/DD RAM volgt uit vorige instructie
- voor het uitvoeren van deze instructie MOET een CG of een DD RAM adres set zijn uitgevoerd, behalve als de laatste instructie cursor shift was.
- het adres wordt automatisch verlaagd/verhoogd volgens entry mode.
- er treedt geen display shift op ongeacht de entry mode.

Het zelf definiëren van karakters is door mij nog niet uitgeprobeerd, dus hierover kan ik weinig vertellen. Verder kunnen ongebruikte geheugenplaatsen bij een 16 of 20 karakter display gebruikt worden voor eigen dataopslag, de controller op het display weet hoeveel karakters er per regel zijn en zal de niet gebruikte geheugenplaatsen niet veranderen tijdens het shiften of zal deze niet adresseren bij het veranderen van de adrescounter. Bijvoorbeeld bij het schrijven van data bij een

display met zestien regels zal de adrescounter als volgt aanpast worden: ...#0C - #0D - #0E - #0F - #40 - #41...
De adressen #10 t/m #27 en #50 t/m #67 zijn dan vrij voor eigen toepassingen.

INITIALISATIE

Bij power-on moet de eerste instructie FUNCTION SET zijn, hierna volgen display on/off controll en entry mode set. Hierna kan je data naar het display gaan schrijven. In de 4 bits mode is dit anders en wie het display in deze mode wil gebruiken moet maar in de datasheet gaan snuffelen. Ik wil hier niet de hele datasheet gaan overtypen.

Als de voedingsspanning slecht opkomt en de interne reset-circuits niet werken moet er een hele initialisatie doorlopen worden. Als dit bij iemand het geval is moet de voeding verbeterd worden of deze moet mij maar bellen voor inzage van de datasheet (of er zelf één regelen natuurlijk).

PROGRAMMAVOORBEELD

Het programma van de voorversterker is gemaakt met een cross-assembler op de PC. Bij deze cross-assembler is het mogelijk om met de instructie FCS tekst op een bepaald adres weg te schrijven en hiervan heb ik dan ook gebruik gemaakt. Als eerste zet ik de lengte van de tekst in de "tekststring". Deze wordt later gebruikt. Verder kan de cross-assembler zelf dingen berekenen, dus laat ik hem het adres waar de tekst staat verdelen in een highbyte en een lowbyte. Hier zal H_CD dus gelijk zijn aan C0 en L_CD gelijk zijn aan 00. Ook deze komen later van pas.

(het teken voor hexadecimaal is hier \$ en niet #. # betekend hier immediate)

```
*      TEKSTEN VOOR DISPLAY VANAF $C000
*      LENGTE TEKST MAX 15 KARAKTERS,
*      IN ASCII; 123456789:;<=>?
```

```
TEKST  EQU      $C000
        ORG      TEKST          *   adres van tekst

T_CD   FCS      /5CD           /   *   lengte tekst + tekst
H_CD   EQU      T_CD/$100      *   berekening highbyte H_CD
L_CD   EQU      (T_CD*$100)/$100 *   berekening lowbyte L_CD

T_TUNER FCS      /5TUNER      /
H_TUNER EQU      T_TUNER/$100
L_TUNER EQU      (T_TUNER*$100)/$100
```

en zo verder tot alle teksten geweest zijn.

In het voorbeeld is de initialisatie niet meegenomen, wel alle declaraties voor het display, die zijn gemaakt voor het begin van het echte programma.

* DISPLAY ADRESSEN

```
D_CMND EQU $B000 * commando register display
D_DATA EQU D_CMND+1 * data register display
```

* GEBRUIKTE GEHEUGENPLAATSEN VOOR OVERDRACHT TEKSTADRES

```
T_ADR_L EQU $00
T_ADR_H EQU T_ADR_L+1
T_ADRES EQU T_ADR_L
```

* DISPLAY COMMANDO'S

```
D_INIT EQU $38 * initialisatie display
D_CLEAR EQU $01 * clear display
D_HOME EQU $02 * cursor home
D_SETCG EQU $40 * set CG ram adres
D_SETDD EQU $80 * set DD ram adres
* ENTRY MODE SET
D_DE_FR EQU $04 * AC=AC-1, geen shift
D_DE_NF EQU $05 * AC=AC-1, display shift mee
D_IN_FR EQU $06 * AC=AC+1, geen shift
D_IN_NF EQU $07 * AC=AC+1, display shift mee
* DISPLAY ON/OFF CONTROL
D_CTRL1 EQU $08
D_CTRL2 EQU $0A
D_CTRL3 EQU $0C
D_CTRL4 EQU $0E
D_CTRL5 EQU $09
D_CTRL6 EQU $0B
D_CTRL7 EQU $0D
D_CTRL8 EQU $0F
* CURSOR/DISPLAY SHIFT
D_CUR_L EQU $10 * cursor shift left
D_CUR_R EQU $14 * cursor shift right
D_DIS_L EQU $18 * display shift left
D_DIS_R EQU $1D * display shift right
```

En dan nu het programma. Het gaat hier om de aanroep van het plaatsen van de tekst. Doordat de plaats op het display en het adres van de tekst voor het aanroepen in de drie registers wordt gezet kan deze aanroep overal in het programma plaatsvinden. Onderstaand programma is in feite het testprogramma wat ik gebruikt heb om deze aanroep uit te testen. In het besturingsprogramma van de versterker zijn deze aanroepen dan verspreid over het hele programma toegevoegd.

```
ORG $8000 * linkadres

JSR BUSY
LDA #D_CLEAR * clear display
STA D_CMND

JSR BUSY
LDA #D_IN_FR * AC=AC+1, display shift niet
STA D_CMND
```

* OPGEVEN TEKST

```
LDY    #$00    * Y = display adres = $00
LDX    #L_CD   * X = $00
LDA    #H_CD   * A = $C0
JSR    DISPLAY
```

```
LDY    #$41    * Y = display adres = $41
LDX    #L_TUNER * X = $07
LDA    #H_TUNER * A = $C0
JSR    DISPLAY
```

```
JSR    BUSY
LDA    #D_CTRL3 * display aan
STA    D_CMND
```

```
RTS
```

* Hierna zal er dus het volgende op het display staan:

```
CD
TUNER
```

* SUBROUTINE DISPLAY; VERZENDEN TEKST NAAR DISPLAY

```
DISPLAY STA    T_ADR_H    * (T_ADR_H) = H_CD
TXA
STA    T_ADR_L    * (T_ADR_L) = L_CD

LDA    (T_ADRES)    * A = lengte tekst
AND    #$0F        * ASCII -> werkelijke lengte
TAX    * X = lengte tekst

TYA    * A = adres eerste letter
ORA    #D_SETDD    * A = A + instructie
JSR    BUSY
STA    D_CMND    * AC = adres eerste letter

LDY    #$01    * Y = 1

NOGTKST LDA    (T_ADRES),Y    * A = karakter tekst
JSR    BUSY
STA    D_DATA    * karakter naar display
INY    * Y = Y + 1
DEX    * X = X - 1
BNE    NOGTKST    * spring als er nog tekst is
ENDTKST RTS
```

* SUBROUTINE BUSY; CONTROLEER OF DISPLAY KAN ONTVANGEN

```
BUSY    PHA    * A naar stack
BEZIG   LDA    D_CMND    * lees status
ASL    A    * zet bit 7 in carry
BCS    BEZIG    * branch als bit 7 hoog is
PLA    * A van stack af
RTS
```

Een programma uitbreiden is nu geen probleem meer, gewoon de tekst achter de andere teksten zetten. Hierbij hoef je zelf niets te berekenen. Vervolgens kun je met drie instructies extra de tekst overal in een programma oproepen.

Bij lange teksten zal het gemakkelijker zijn om met een end-of-line karakter te werken, aangezien de maximale tekstlengte 15 karakters mag bedragen. De lengte wordt namelijk opgegeven in ASCII en een register kan slechts één ASCII waarde bevatten. Verder is een tekst met lengte nul niet echt te gebruiken.

In voorgaande publikaties in Atomnieuws heb ik geschreven dat ik gebruik wilde maken van een fluoricentie display. Deze zijn bij mijn weten echter nergens te koop (of heel duur), zodat ik mijn toevlucht heb genomen tot de LCD displays. Voor de versterker zelf gebruik ik een LTN211 met LED achtergrondverlichting en dit ziet zeer fraai uit.

Een verdere toepassing van een display zal misschien het DTMF modem worden, of misschien een status-display voor de Atom(-in-pc). Niet zoals vroeger een circus van LED's op de atom, of daarna een 7-segments LED-display met daarop het nummer van de voorgeschakelde box, maar een display met daarop bijvoorbeeld de tekst

```
BOX = PCHARME  
WRITEPROTECT=OFF
```

Dit is alles slechts een kwestie van software.....

Ik hoop dat nu duidelijk geworden is dat het aansturen van zo'n display geen moeilijke zaak is en dat deze informatie kan worden gebruikt voor het vinden van eigen toepassingen en het schrijven van eigen basic of assembly programma's. Als alles goed gaat, zal ik een tweetal displays demonstreren op de landdag van 8 oktober. (als ik het weekend vrij krijg van het opperbevel van de Nederlandse Landmacht, sectie verlofzaken, of zoiets.)

Het tweede display is aangesloten op het 6532 board, waarop echter, door problemen met de GAL, geen GAL en 6532 meer zitten. Slechts EPROM, 6502 en 7410, dus totaal geen RAM en toch loopt er een uitgebreid demoprogramma! Komt dat zien.

Guido Baltus
B. Lünenschloszstraat 8
6137 PJ Sittard
046 - 521035 (na oktober 1994: 046 - 4521035)

Literatuurlijst:

- Product specification LTN211 Liquid Crystal Display
july 1990 Philips Components
- Datasheet NJU6408B LCD controller/driver
New Japan Radio Co., LTD

WinAtom, de stand van zaken

Natuurlijk kent u inmiddels allemaal de terminalemulator van Roland. Deze werkt perfect in DOS en ook in een DOS box onder Windows.

Zoals u wellicht al eens ter ore is gekomen, ben ik bezig met een zelfde soort terminal emulator, maar dan onder Windows. Bij deze een verhaaltje over de vorderingen van dit project.

Allereerst: het besturen van I/O.

Onder Windows mag je niet zondermeer een I/O instructie geven. Dit komt omdat Windows een schil is rondom de hardware en de programmatuur en (dus) niet toestaat om direct de hardware aan te spreken. Windows schuift naar eigen goeddunken met intern geheugen en zo kan het zijn dat wat het ene moment nog valide is, het volgende moment door een ander programma wordt gebruikt.

Maar uiteraard is de Windows kernel uitbreidbaar, zodat ook de hardware kan worden aangesproken. Denkt u maar eens aan een commerciële uitbreidingskaart alwaar keurig de .DRV en .386 bestanden worden bijgeleverd. Ik heb getracht om ook een .386 driver te bouwen, maar deze levert om nog niet doorgrondde redenen af en toe problemen op. Vandaar dan ik naar een ander middel heb gegrepen: de DLL.

Wat is een DLL

Voor de niet Windowieden onder ons: een DLL is een dynamische bibliotheek, die geladen wordt op het moment dat het nodig is en weer vrij gegeven wordt op het moment dat het programma beëindigd wordt. Tot zover eigenlijk niet interessant edoch.. bij de aanmaak van een DLL kan je opgeven dat het geheugenbeslag FIXED is, wat zoveel wil zeggen, dat ook de Windows kernel er met zijn fijnbesnaarde handjes van afblijft. Van deze eigenschap wordt door bovengenoemde DLL uitbundig gebruik gemaakt.

De DLL is een uiterst kleine DLL welke uitsluitend met de hardware communiceert. De rest gebeurt in de applicatiesoftware. ATOM.DLL heet het genoemde, en moet óf in de directory staan waar ook de WinAtom programmatuur staat, óf in de systeem directory van Windows. Aan u de keuze.

Er is geen installatieprocedure nodig om deze DLL aan Windows kenbaar te maken. De applicatieprogrammatuur zorgt voor het laden van deze DLL.

En toen.. Een atom lettertype

Het zal u nauwelijks verbazen dat de Atom zijn eigen karakter-set heeft. Met name de fraaie 'clear 0' blokjes alsmede de geïnverteerde uitvoeringen van de hoofdletters maken de Atom tot dat wat hij is. Dat moet dus ook zo onder Windows.

Windows kent 2 soorten lettertypes (eigenlijk meer, maar dat is voor dit verhaal niet interessant): de bitmap en de TrueType fonts. Bitmap font kent u, misschien ook zonder dat u het weet, allemaal. Een matrix printer bouwt zijn regel op, op basis van een aantal puntjes. Bij een bitmap is dat net zo. Er is echter een probleem: Voor iedere soort kaart dient een overeenkomstig font te zijn. Maak je een font van, zeg 16x32 puntjes, dan ziet dat er op een VGA kaart heel anders uit dan een Supervga kaart. In het laatste geval lijkt de Atom dan meer op een postzegel. Hercules kaarten, de oude monochrome kaarten, doen helemaal merkwaardige dingen met zo'n font ofwel: dat kan ik de Atomclubleden niet aandoen.

TrueType

TrueType is een verzamelnaam voor fonts, lettertypes, waar de eigenschappen van worden vastgelegd en op het moment van gebruik deze worden omgerekend naar fysieke eigenschappen. Met andere woorden: de letter O wordt beschreven als: twee cirkels met onderling een vastgestelde afstand. Tijdens het bepalen van de grootte hoeft er alleen vermenigvuldigd te worden om een fraai uitgevoerde O te krijgen. En nou maar niet denken dat dat zondermeer eenvoudig in elkaar te zetten is. Nee, plotseling moet je de taal van een ervaren drukker doorgronden, met begrippen als 'kerning' en 'hinting'. Maar goed, we doen net alsof we alle vrije tijd van de wereld hebben en dus is het Atom TrueTypefont een feit.

Printen van een TrueType font

Het mooie van TrueType is dat dit zondermeer naar een printer kan worden gestuurd. Uiteraard stelt dit wel wat eisen aan de

printer, deze moet immers graphics ondersteunen. Dat moet inmiddels niet zo'n probleem meer zijn, tenzij u nog een omgebouwde telex aan uw PC heeft hangen. In geval het mogelijk is, maakt Windows er een Softfont formaat van voor de printer. Op een HP laserjet III ziet het er allemaal erg gelikt uit. Voorlopig heb ik er voor gekozen om een Clear 0 scherm op volledige papierbreedte af te drukken. Het is instelbaar of dit portrait of landscape moet worden. Wellicht ooit in een latere versie instelbaar.

Op een van de volgende pagina's vindt u een overzicht van de lettertypes. (ZIE LOSSE BIJLAGE)

En hoever is het nu

Nou, dat valt eigenlijk best tegen. Dat smoesje van druk druk druk, kent u ongetwijfeld, maar er zijn soms activiteiten die je nou eenmaal van de Atom in PC afhouden. Moet kunnen toch? Maar goed. De stand van zaken. De terminal emulatie begint er qua front-end aardig op te lijken. De communicatie tussen Windows en de Roland-kaart lopen goed. Clear 0 en alles wat daar bij hoort lijkt redelijk te verlopen en het printen van dit soort zaken gaat ook naar wens.

Clipboard

Ondersteuning voor het clipboard, klembord in slecht vertaald Nederlands, is er ook. Het is mogelijk om met de muis en linkermuistoets een selectie van het Atom scherm over te hevelen naar het clipboard. Met een optie valt te kiezen of dit in ASCII formaat is, of in een bitmap formaat. In het eerste geval valt er nog iets zinnigs te doen met de informatie, in het tweede geval is het een 'plaatje' en kan u het in Word of iets anders als plaatje 'paste-en'. Eventueel ook nog inlezen in Painbrush of Paintshop en er allerlei enge dingen mee doen. 'The PC is the limit, not the Atom' weet u nog. Verder is er ook aan DDE support gedacht, maar echt zinnig werkt dit nog niet.

DDE, Dynamic Data Exchange

Het is nu mogelijk om een stuk van het scherm aan te geven en als dit wijzigt, wordt dit automatisch (via een warm-link,

voor de kenners) door gestuurd naar de applicatie die dit aanvraagt.

Feitelijk is DDE niet echt zinvol. Beter zou zijn om het middels een OLE verbinding te laten lopen, zodat het mogelijk is om bijvoorbeeld vanuit een Excel spreadsheet de Atom commando's te geven.

Maar OLE en met name OLE 2 is niet echt eenvoudig. Ik daar een boekje voor gekocht hebben. Ik daar echter nog steeds niet alles van begrijpen. Ik dus ernstig zorgen hebben of er ooit een OLE 2 koppeling zal komen. Maar goed. Wie weet. De tijd zal het leren.

En wat nog meer

Verder moeten er nog een aantal dingen worden uitgedacht. Niet in de zin van 'hoe moet dat' maar meer in de zin van 'hoe doe je dat handig'. Bijvoorbeeld het laden van een file. Is het nou echt handig om dit via een Windows dialog box, u weet wel zo'n box die je krijgt op het moment dat je een file opent sluit etc., of is het toch handiger om het op de commandline in te tikken. Voor beide valt wat te zeggen, maar wie het echt weet mag het zeggen of me bellen.

Een ander moeilijkheid zijn de grafische modes. In principe kan je dit als een Windows Bitmap beschouwen, maar als je voor het updaten van iedere punt door de Atom iedere keer een nieuwe bitmap moet tonen, dan wordt je geplaagd door (het gebrek aan) performance. Voorlopig heb ik het opgelost door het in stukken knippen van de bitmaps. Een soort 'banding' dus. Het printen van zo'n bitmap is overigens weer een fluitje van een dubbeltje (sorry, de cent is uit de mode), want hierin wordt standaard voorzien door de Windowskernel.

Verder is er overleg geweest met Roland over hoe een en ander slim aan te pakken. We hebben nog gespeeld met de gedachte om het volledige terminal gebeuren over te zetten naar C, maar de twee terminalemulators zijn in basis dermate anders dat dit wel leuk, maar niet praktisch is. Maar gezien de opzet van de Atom in PC, nogmaals hulde Roland, is het feitelijk eenvoudig om beide terminalprogramma's op basis van functionaliteit in de pas te laten houden. De hoop is alleen dat het tempo van Roland is bij te houden, want het zal altijd zo zijn dat de Windows versie achterloopt op die van DOS.

En nu verder

Met een beetje geluk valt er op de komende Atomdag iets te bewonderen onder Windows. In ieder geval de basis functionaliteit en als er wat voortgang is geboekt met de bitmap omgeving dan dat ook. Het produkt is zeker nog niet voor distributie beschikbaar, domweg omdat het nog niet af is. Verder is het zo dat ik eigenlijk erg kritisch ben op eigen werk, dus als ik iets beschikbaar stel, dan mot 't wel werke.

Kortom: het is een project met een behoorlijke omvang aan het worden. Het zal toch maar je hobby zijn. Ik houd u op de hoogte. Ook van de eventuele ontwikkeling van de Soft Atom van Dick Bronsdijk onder Windows. Maar daar is eigenlijk nog niets aan gebeurd.

P.S. Wie heeft er overigens een programmaatje om vrije tijd te genereren?

ACORN ATOM USERS GROEP DEN HAAG

Geachte Atomdiscipelen,

De aanstaande bijeenkomst op 28 oktober zal niet meer bij Louis van Meerloo gehouden worden. Louis gaat namelijk op korte termijn verhuizen naar buiten de regio.

Dientengevolge zullen de volgende avonden gehouden worden bij:

Theo Waayer
Hendrik van Boeijenlaan 66
2273 DC Voorburg
tel. 070 3862504.

De eerste avond bij Theo Waaijer op 28-10 begint om ca. 20.30 u.

Verder wordt er in de wandelgangen gesproken over , voorlopig regionale , aansluiting bij de BBC-club. In deze club is het gebruik van "Acorn-PC's" een stuk verder doorgedrongen dan bij ons, maar men wil daar een "8-bits hoek " gaan oprichten.

Misschien is het voor beide clubs een nieuwe impuls weer wat meer leden op de regioavonden te zien.

De regioavonden zullen, als deze aansluiting zou doorgaan, op donderdag-avond gaan vallen. Graag zou ik willen weten hoe jullie tegenover dit idee staan. Je kunt reageren per telefoon of met een briefje.

Vriendelijke groeten, Theo Waayer.

BIG-BEN LANDDAG - 1 OCTOBER a.s.

Pieter Groen college , Helmburgweg , Katwijk a. Zee.

** Wie van onze ATOM-in-PC gebruikers is daar aanwezig ? **

O JEE,

KOPPELING VAN ELECTRON EN PC

door roland leurs

Het "gerucht" gaat al een tijdje door de Limburgse Atom Club dat ik ooit eens een Electron aan een PC ga koppelen. Ik heb het op een jaarvergadering eens voorgesteld als activiteit waarmee ik me eens zou gaan bezighouden. En dat heb ik gedaan

De Electron heeft met zijn relatief weinig geheugen en (naar huidige maatstaven) matige grafische resolutie toch iets bijzonders. Net als vroeger bij de Atom, daar hadden we Snapper en Space Invaders. Spellen die je gewoon waardeert omdat ze zo weinig geheugen gebruiken en toch zorg(d)en voor uren speel plezier. De Electron is net zo'n machine. Neem eens spellen als Repton, The Way of the Exploding Fist of ook hier Snapper. Stuk voor stuk draaien ze toch op een machine met maximaal 32 kb geheugen.

Een nadeel is dat bij de meesten onder ons dat geheugen gevuld wordt vanaf cassette.

Zo ook bij ons thuis. Pascalle klaagt steeds dat het laden lang duurt. (Ja, want soms werk ik met de 486 en VGA-kaart.) Dus wanneer komt er een disk-drive voor de Electron. Waarschijnlijk zal dat nog lang duren. Maar, wie heeft er nou een disk-drive nodig voor de Electron als daar verschillende PC's met veel harddisk in huis staan? Niemand toch!

En daar begint een nieuw verhaal waarvan de titel boven aan deze bladzijde staat. Jawel lieve lezers, het is gebeurd. De Electron en de PC hebben hun bussen aan elkaar geknoopt. (Even tellen, en dat is al de tweede 6502 aan mijn PC). Wat is er nu weer uitgevonden? Niet veel, wel spectaculair.

Ik ben uitgegaan van de buskoppeling tus'sen Electron en Atom. U weet wel, de print uit AN11-3. Werkt prima dus hardware aan Electron zijde is al af. In de PC zit een I/O kaartje met een 8255, hardware ook aan deze kant af; althans dat hoopte ik. Het bleek niet waar te zijn. Beide communiceerden wel maar niet goed genoeg. Sommige bytes kwamen heel veel aan, andere nooit. En omdat er een bidirectionele databus in gebruik is, moest de 8255 zijn A-poort schakelen tussen ingang en uitgang, hetgeen altijd extra data opleverde (?).

Daar moest nog wat aan gesleuteld worden. De I/O print werd verbouwd. Er kwam wat extra adressering bij en ziedaar: de aansturing voor de diverse registers is gerealiseerd. Het

werkt nu wel, maar over een definitieve uitvoering voor de rest van de wereld (u dus) moet ik nog even nadenken. U hoort er nog van.

Vervolgens uiteraard een draad leggen van Electron naar PC en de verbinding was een feit. Dat de PC niet wilde starten kwam in eerste instantie doordat de I/O kaart verkeerd om in de bus geprikt was (slecht vier IC's kapot) en daarna omdat de Electron nog uit stond. Maar ja, iedereen heeft wel eens een dag dat niet alles goed gaat en hobby's moeten nu eenmaal geld kosten. Op welke manier dan ook.

Bij hardware hoort ook software. De verbinding bestaat wel, maar er moet natuurlijk ook gecommuniceerd worden. Hoe pak je zoiets aan? Heel simpel, zeker als je ooit eens een terminal programma voor een Atom-in-PC-kaart geschreven hebt. Dan heb je op je PC al een pakket dat commando's kan ontvangen en er dan iets nuttigs mee kunt doen. Alleen even de communicatie routines aanpassen en de software is al klaar aan PC zijde. Met deze oplossing krijg je dus via de koppeling een tweede monitor aan je Electron, een muis, een printer, joystick, (hard)disk en een Atom. (Tuurlijk want alle hardware op de PC-bus staat tot je beschikking, dus ook de Atom-in-PC kaart. Hebben we meteen weer een koppeling Atom-Electron.

Voor de Electron worden nu wat softes waren geschreven worden om via de regels van het operating systeem een Disk Filing System op te zetten. Bestudering van de 'Advanced Electron User Manual' leert dat het betrekkelijk eenvoudig is om met de beschikbare PC-commando's een Acorn compatibel DFS te schrijven. Ik hoop dit op de eerstvolgende landdag te demonstreren.

Hoe zat het ook alweer in elkaar? Even terugblikken naar Atom Nieuws 11-4.

De hardware voor de PC bestaat slechts uit de gebruikte registers en de adressering daarvan. De hardware voor de Electron bestaat eveneens uit de gebruikte registers met bijbehorende adressering, maar verder ook nog uit een EPROM met de bijbehorende aansturing.

De genoemde registers zijn een 'transmit data register' (TDR), een 'receive data register' (RDR) en een status register (STR).

Om data van de ene naar de andere computer te zenden dient naar het TDR geschreven te worden. Als data ontvangen wordt, dan moet deze uit het RDR gelezen worden. In het STR wordt aangegeven of data verzonden resp ontvangen zijn. Wat gebruik betreft lijkt dit op een normale seriële poort.

Als een van de computers naar het TDR schrijft, wekt de adres-

decodering een actief lage puls op; dit is hetzelfde als een chipselect signaal voor een geheugenchip. Bij de opgaande flank van die puls wordt de data van de databus in het register geklokt. Tegelijkertijd wordt in de bijbehorende flipflop van het statusregister een logische '1' geklokt. De toestand van deze flipflops kan door beide computers uitgelezen worden en wel bit 2 in het statusregister van de zendende computer en bit 0 in het statusregister van de ontvangende computer.

Bij het uitlezen van het RDR wordt de ingang OC (output control) van het betreffende register laag gemaakt. Hierdoor komt de data beschikbaar op de databus van de betreffende computer, dit is te vergelijken met het lezen uit een geheugenchip. Gelijk met het uitlezen van een RDR wordt ook de CLR-ingang van de betreffende flipflop laag gemaakt waardoor de flipflop gereset wordt. De uitgang van de flipflop en dus ook de bovengenoemde bits uit het statusregister worden dan logisch '0'.

Voor de software houdt dat in dat er gewerkt kan worden met een betrouwbare handshake tijdens de datacommunicatie. De zendende computer test eerst aan de hand van bit 2 van zijn statusregister of de andere computer gereed is om data te ontvangen. Dit is het geval als dat bit '0' is. Dan pas mag de nieuwe data in het TDR geschreven worden. De ontvangende computer hoeft alleen maar naar bit 0 te kijken of er data gezonden is. Dat is het geval als bit 0 de waarde '1' heeft.

De Electron heeft nog een extra register: het 'bankselect register' voor de EPROM. Dit is noodzakelijk omdat de EPROM op een "schakelkaart-achtige" wijze in het geheugen van de Electron is geplaatst. Deze EPROM staat in het gebied &FC00 t/m &FDED, verdeeld in 32 banken van 510 bytes. Als we een bepaalde positie in de EPROM willen lezen dan moet er eerst voor gezorgd worden dat de juiste bank van de EPROM voorgeschakeld staat. Hiertoe dient het bankselect register (BSR).

Tot zover de belangrijkste geheugenopfrissing. De complete uitleg staat in AN11-4 vanaf pagina 15.

Zo ziet u dat mijn matige vakantieweer toch nog zijn zonnige kanten heeft, met vriendelijke groeten,
Roland Leurs

Of er nog een Electron-in-PC verschijnt? Voorlopig niet, maar je weet maar nooit

Het programmeren van de 65802

Na succesvolle uitvoer van de operatie aan de Atom, zou het zo kunnen zijn dat ook u inmiddels een echte SxAtom heeft. Als dat zo is, is dit artikel speciaal bedoeld voor u.

Wat gaan we doen. Wel allereerst het bespreken van het programming model. Een saai maar noodzakelijk stukje kwaad. Het is net als de toonladders oefenen bij het leren piano spelen.

De power-on status

Als de 65802 wordt opgestart, dan start deze in de emulatiemode op. Dit heeft tot belangrijk voordeel, dat de standaard 6502 opstart procedure kan worden gevolgd. De timing van alle instructies is conform de originele NMOS 6502. In tegenstelling tot de NMOS 6502 is het echter zo dan de 'illegale opcodes', zeg maar de niet gedefinieerde instructies, binnen de 802 weldegelijk een betekenis hebben. Dus de niet gedocumenteerde instructies van een 6502: vergeet ze maar.

Veel valt er nog te zeggen over de emulatiemode, maar natuurlijk is het veel interessanter om over de echte 16 bit mode te praten!

De 65802 in de native mode

Er zijn een paar nieuwe registers beschikbaar. In volgorde:

Het Direct Page register.

Het DP register is een aanvulling op de zero page zoals die bekend is. Weet u nog die LDA (#5A), Y instructies? En het gebrek aan zero-page adressen? Nou dan zal het nut van het DP register spoedig duidelijk worden. In plaats van Zero page adressering, is het nu mogelijk om op elke page in het geheugen, een LDA (#5A), Y te doen. Wordt het DP register ingesteld op 0, dan blijft het zoals we dat kennen van de echte 6502, wordt het op bijvoorbeeld #27 gezet, dan is er een volledige reeks van 255 'Zero page' (hoewel die niet zo heten), beschikbaar. U raadt het al: bij opstarten wordt dit register op 0 gezet.

De Stackpointer

In de, native mode bestaat de stackpointer uit een 16 bits adres. Dus niet langer uit acht bits zoals we dat kenden, met

als consequentie dat alleen op page 01 stackruimte is. Voordeel is tevens dat de stackgrootte niet langer tot 255 gelimiteerd is, hoewel ik eerlijkheidshalve moet zeggen dat ik nog nooit tegen die limiet ben aangelopen.

Voordeel van deze stackpointer kan zijn, is dat het mogelijk is binnen een eigen routine een eigen stack te maken. Kan handig zijn bij bijvoorbeeld recursief programmeren. Betekent in de praktijk dus de volgende stappen: stackpointer bewaren; nieuw gebied uitzoeken voor stackpointer; doen wat nodig is; bewaarde stackpointer terug halen en klaar.

De Accumulator en Indexregisters (A, X, Y)

Het grote verschil tussen de 6502 en de 802 is het gebruik van 16 bit registers voor A, X en Y alsmede de mogelijkheid om tussen 8 en 16 bits precisie te switchen. Dit alleen al is geweldig: bijvoorbeeld bij het aftrekken van twee 16 bits getallen. In plaats van minimaal 6 instructies, is het nu mogelijk om het in 3 instructies te doen.

De breedte van de registers (8 of 16 bits) kan worden ingesteld met 2 bits uit het status register, het index register select (x) en memory/accumulator select (m). Indien beide flags gezet zijn, dan is dat de status zoals we die kennen van de 6502. In onderstaande tabel is een opsomming van de mogelijkheden gegeven.

m bit aan	8 bit accu
x bit aan	8 bit index register
m bit aan	8 bit accu
x bit uit	16 bit index register
m bit uit	16 bit accu
x bit aan	8 bit index register
m bit uit	16 bit accu
x bit uit	16 bit index register

Afhankelijk van de status van deze flags, voert de 802 een andere instructie uit bij de volgende opdracht. Kijk maar even mee:

```
BD 00 B0          LDA #b800, X
```

Deze instructie laadt de accu met de waarde van (#b000+x). Het X register kan een waarde hebben tussen 0-255 of 0-65535,

afhankelijk van de 8 of 16 bits flag. Verder kan er een 16 bit getal of een 8 bits getal in de accu worden geladen, ook weer afhankelijk van de m flag. Even opletten dus.

De opcode BD, die dus staat voor LDA (#xxxx), X, vindt u op alle processoren terug, het gedrag wordt bepaald door de stand van de flags.

Het schakelen tussen 8 en 16 bits

De twee genoemde register select flags worden gezet of geschoond door een speciaal in het leven geroepen instructie om de individuele flags in het P of Status register te zetten. De instructies heten SEP en REP, wat staat voor 'SEt P flag' en 'REset P flag'. Als operand wordt de flag opgegeven waar het om gaat. Het maakt trouwens niet uit welke flag gezet wordt. De volgende instructies zijn dus elkaars equivalent:

SEC	Set Carry Flag
SEP	@#01

Let erop dat de Carry staat op bit 1 van het P of Status register. Vandaar die #01 als operand.

Het schakelen van en naar 16 bit is natuurlijk leuk een aardig, maar wat gebeurt er nou precies. Het is namelijk van belang om te weten wat er gebeurt. In onderstaande afbeelding wordt een en ander wat toegelicht.

Resultaat van het schakelen van registergrootte
L = bits in laag byte; H = bit in hoog byte

Index Register: 16 bit naar 8			
HHHH HHHH	LLLL LLLL	geeft	0000 0000 LLLL LLLL
flag x=0			flag x=1

Index Register: 8 naar 16 bits			
0000 0000	LLLL LLLL	geeft	0000 0000 LLLL LLLL
flag x=1			flag x=0

Accumulator: 16 bits naar 8

A		B	A
HHHH HHHH LLLL LLLL	geeft	HHHH HHHH LLLL LLLL	
flag m=0		flag m=0	
-----Ook C-----			

Accumulator: 8 naar 16 bits

B	A		A
HHHH HHHH LLLL LLLL	geeft	HHHH HHHH LLLL LLLL	
flag m=1		flag m=0	

U ziet het: Als een van de index registers X of Y wordt omgeschakeld van 16 bit naar 8 bits, dan gaat het resultaat van het hoogste byte verloren. Logisch.

Andersom is het zo, dat als van 8 naar 16 bit wordt omgeschakeld, het hoogste byte met 0 wordt gevuld. Ook weer volkomen logisch.

Iets anders gaat het met de Accu. Als voor de accu de 8 bits flag aanstaat, is er een verborgen B accu, die de waarde van het hoge byte van de 16 bit notatie bevat. Deze B accu mag gezien worden als een appendix van de A accu. De grap hiervan is, dat het schakelen van en naar 16 bit geen verlies aan gegeven betekent.

In de officiële notatie heet het zo te zijn dat in geval van 8 bit mode de accu's A en B heten, in geval van de 16 bit mode worden A en B samengevoegd en heet de accu C. Deze bestaat dan dus uit A als low byte en, jawel, B als high byte.

Tot zover deze keer. Toch nog wat wat huiswerk.

1. Laat u mij eens weten of dit onderwerp uw interesse heeft. Dan weet ik of het gelezen wordt en of er werkelijk belangstelling voor is.
2. Wie heeft er, of wil deze aanpassing maken aan de step routine in de Jbox, zodat ook de 802 instructies kunnen worden gestept?

Programmeertip: files zoeken in een directory

door roland leurs

U kent het programma vast nog wel uit AN10-3 waarbij een PC programma een Atom clear 4 dump van disk laadt en vervolgens op het scherm toont.

Aangezien iedereen vast wel een diskette heeft met daarop een aantal AGS files is onderstaand programma wel een aardige demo.

```
10 REM GRAPHICS SHOW voor Atom in PC
20 REM DOOR ROLAND LEURS
30 *FIND A:*.AGS
40 CLEAR 4
50 IF ?#140 THEN TXMOD;EXIT
60 OSCLI "LOAD A:"$#14A" 8010", $13
70 FILL #8000, #801F, #00
80 PAUSE 120
90 *FIND
100 GOTO 50
```

Het programma demonstreert hoe eenvoudig de Atom-in-PC een directory kan doorzoeken naar files die aan een bepaalde filespecificatie voldoen. In regel 30 wordt de eerste zoekactie gedaan. Hierbij wordt opgegeven aan welke specificatie de files moeten voldoen (in dit geval alle files met extensie AGS). Als een file gevonden is die aan de specificatie voldoet dan bevat adres #140 de waarde #00. Vanaf adres #14A staat dan de filenaam als normale ascii string.

In regel 90 wordt een volgende file gezocht volgens dezelfde specificatie. Ook hier geldt weer dat adres #140 een #00 bevat als er nog een file gevonden is. Zie voor meer informatie omtrent *FIND in de handleiding van de Atom-in-PC of type

HELP *FIND

vanaf de Atom prompt. U krijgt dan meer informatie over dit PC-DOS commando.

Met de vriendelijke groeten van Roland !

VERSIENUMMERS

Daar Roland Leurs erg intensief bezig blijft met het verder ontwikkelen van Hard- en Software voor de Atom-In-PC-kaart, zou er wel eens verwarring kunnen ontstaan over de meest recente versienummers van operatingsystemen en terminalprogramma's. Iedereen blijft graag op de hoogte van de laatste ontwikkelingen terzake en daarom ben ik maar eens gaan snuffelen in Atom-Nieuws naar wat bruikbare routines.

In het programma "RAD VAN FORTUIN" op de laatste regio-schijf wordt er al op sommige dingen getest om het programma goed te kunnen draaien. Deze routines heb ik dan maar gebruikt in het programma'tje "VERSIENUMMERS".

Om alles correct te laten verlopen, dient de ATOM-IN-PC-kaart als volgt te zijn opgestart :

```
ATOM /R=ESC /V=VGA /F=C:\ATOM\FONT\ATOMVDU.FNT
```

Uiteraard komt in de plaats van /V=VGA uw eigen videokaart.

Na het initialiseren van de ATOM, met *INIT, kan het programma VERSION gestart worden en moet het resultaat er ongeveer uitzien zoals hetgeen onder de listing te zien is.

Omdat ik met mijn "OUDE" Atom's niet altijd zeker was of de battery-backup wel goed werkte, had ik een programma'tje dat de checksum van elke box alsook van het geheugen berekende. Dit programma vond ik ook wel interessant en nuttig voor de Atom-in-pc kaart en heb dit dus ook maar daarvoor aangepast. Ook de hierin gebruikte routines zijn "geleend" van meer bekende voorgangers uit onze club. (Ere wie ere toekomt aan Leendert Bijnagte en anderen.)

Om een lang verhaal kort te maken, het resultaat na het draaien van het programma zou er min of meer moeten uitzien zoals bijgevoegde lijst. Als het operatingsysteem in RAM zit, dan hoort achter "SCHAK.SOFT" te staan : #C813. Bij mij draait alles in ROM, vandaar.

Om het systeem in rom te draaien, dient in de schakelsoft op adres #1FAB de waarde #60 te staan en kan met het programma "START" ge-initialiseerd worden. Hoe dat gaat heeft Roland al eens ergens beschreven, maar voor de duidelijkheid dan nog maar eens.

Vanuit DOS de Atom opstarten met :

```
ATOM /R=ESC /V=VGA /F=C:\ATOM\FONT\ATOMVDU.FNT
```

daarna met *INIT het operatingsysteem + de toolboxen laden, vervolgens writeprotect OFF zetten : WP OFF

dan op adres #1FAB=#60 poken, ?#1FAB=#60 waarna de schakelsoft weer op slot gaat met WP ON en dan niet vergeten weg te schrijven met *SAVE"C:\ATOM\SYSTEM\SOSPCWT.ROM"1000 1FFF

Als hierna de ATOM opnieuw opgestart wordt vanuit DOS, kan er gekozen worden tussen het operatingsysteem in RAM of ROM, door het programma *INIT voor RAM ofwel *START voor ROM te kiezen. Hopelijk heeft dit voor sommige leden alles iets duidelijker gemaakt of is het nu nog donkerder?

```

10 PROGRAM VERSIENUMMERS
20
30 FUNCTION VIDEOKAART
40  ZB(0);ZB(#27)
50  VIDEOKAART=((LB+LB*256)/#1000)&3
60 FEND
70
80 FUNCTION LB,A,X,Y
90  USR #FFC5
100 LB=A
110 FEND
120
130 FUNCTION VERSION
140  ZB(0);ZB(#46)
150  VERSION=LB+100*LB
160 FEND
170
180 PROC ZB(A)
190  LINK #FFC8
200 PEND
210
220 ESCAPE=OFF
230 ZERO
240 @=0
250 P.$12"Er zit een ";O=VIDEOKAART
260 IF O=0;P."CGA -"
270 IF O=1;P."EGA -"
280 IF O=2;P."VGA -"
290 IF O=3;P."Hercules -"
300 P." kaart in Uw computer"
310 P."Uw operatingsysteem heeft het volgende versienummer "
320 FOR N=0 TO 3;P.$(N?#FA80);N.
330 P.'"Het terminalprogramma heeft versienummer ".VERSION'
340 ESCAPE=ON
350
360 END

```

```

>Er zit een CGA - kaart in Uw computer
Uw operatingsysteem heeft het volgende versienummer 3.09
Het terminalprogramma heeft versienummer 220

```

```

Er zit een VGA - kaart in Uw computer
Uw operatingsysteem heeft het volgende versienummer 3.09
Het terminalprogramma heeft versienummer 220

```

```
10 REM TEST ACORN-ATOM
20 NONMI
30 P.$12"TEST ACORN-ATOM"$21
40 DIM LL7,AA(12),BB(12),CC(12)
50 F.N=0 TO 12;DIM D(12),E(3),F(5);AA(N)-D;BB(N)-E;CC(N)-F;N.
60 RESTORE 440
70 F.N=0 TO 12;READ $AA(N),BB(N),$CC(N);N.
80 FOR N=0 TO 7;LL(N)--1;N.
90 FOR N=0 TO 1;P-#2800;[
100:LL0 LDY@0
110:LL1 LDA(#80),Y;STA#82;LDA@#EC;STA(#80),Y;LDA(#80),Y
120 CMP@#EC;BNE LL3
130:LL2 LDA#82;STA(#80),Y;INY;BNE LL1;LDA@0;STA#80;RTS
140:LL3 LDA@1;STA#80;RTS
150:LL4 LDA @*10;STA#32;LDY @*00;STY#30;STY#3A;STY#3B
160:LL5 LDA(#30),Y;LDX @*08
170:LL6 LSR A;ROL#3A;ROL#3B;BCC LL7;PHA;LDA#3A;EOR @*2D;STA#3A;PLA
180:LL7 DEX;BNE LL6;INY;BNE LL5;INC#31;DEC#32;BNE LL5;RTS
190];N.;P.$6;@=0;Z=0
200 !#90-#ABCDEF01;P."ZERO PAGE: ";IF!#90-#ABCDEF01;P."O.K.""
210 IF!#90<>#ABCDEF01;P."VERMINKT"
220 P-#1000
230 DO !#80=P;LINK LL0;P=P+#100;U.?#80=0
240 P."#&P-#100" - #"
250 DO !#80=P;LINK LL0;P=P+#100;U.?#80=1 OR P-#2800
260 P.&P-1" RAM O.K.""
270 P."#2800 - #32FF DIT PROGRAMMA"
280 P-#3300;DO !#80=P;LINK LL0;P=P+#100;U.?#80=1
290 P."#3300 - #&P-#101" RAM O.K.""
300 P." ROM / BEREKENDE WERKELIJKE"
310 P." EPROM SIGNATURE SIGNATURE"
320 F.N=0 TO 12;IF N>4;CHIP (N-5)
330 ?#31=BB(N)
340 LINK LL4
350 C=?#3B*#100+?#3A
360 P.$AA(N)," #";IF C<#1000;P."0"
370 P.&C," ",$CC(N)'
380 IF N=4 THEN P." A-BLOKKEN"
390 NEXT N
400 REM DATA "SOORT ROM",ADRES,"SIGNATURE"
410 REM DATA EVENTUEEL AANPASSEN
420 REM AAN EIGEN SYSTEEM
430 REM VANAF REGEL 490 (A-BLOKKEN)
440 DATA "BASIC LOW ",#C0,"#E802"
450 DATA "F.P.ROM ",#D0,"#8D61"
460 DATA "DOS ROM ",#E0,"#13EF"
470 DATA "BASIC HIGH ",#F0,"#8055"
480 DATA "SCHAK.SOFT ",#10,"#C84B"
490 DATA "P-CHARME ",#A0,"#5534"
500 DATA "JBOX 2.1 ",#A0,"#0B7E"
510 DATA "SALFA 2.6 ",#A0,"#9F75"
520 DATA "PCBOX ",#A0,"#1A75"
530 DATA "GAGS 2.4 ",#A0,"#9F4C"
540 DATA "JOSBOX ",#A0,"#D4EA"
550 DATA "PC-UTIL ",#A0,"#B8B6"
560 DATA "LEEG ",#A0," ??? "
570 END
```

```

ZERO PAGE: O.K.
#2000 - #27FF RAM O.K.
#2800 - #32FF DIT PROGRAMMA
#3300 - #9FFF RAM O.K.
  ROM /      BEREKENDE   WERKELIJKE
  EPROM      SIGNATURE   SIGNATURE
BASIC LOW   #E802        #E802
F.P.ROM     #8D61        #8D61
DOS ROM     #13EF        #13EF
BASIC HIGH  #8055        #8055
SCHAK.SOFT  #C84B        #C84B
  A-BLOKKEN
P-CHARME    #5534        #5534
JBOX 2.1    #0B7E        #0B7E
SALFA 2.6   #9F75        #9F75
PCBOX       #1A75        #1A75
GAGS 2.4    #9F4C        #9F4C
JOSBOX      #D4EA        #D4EA
PC-UTIL     #B8B6        #B8B6
GEEN ROM AANWEZIG
LEEG        #DC0B        ???

```

```

10 REM START ATOM-IN-PC-KAART
20 ?#B888=0
30 *LOAD C:\ATOM\SYSTEM\P-CHARME.ROM A000
40 PRINT "P-CHARME ..... loaded at rom 0"
50 ?#B888=1
60 *LOAD C:\ATOM\SYSTEM\JBOX2_1.ROM A000
70 PRINT "JBOX V2.1 ..... loaded at rom 1"
80 ?#B888=2
90 *LOAD C:\ATOM\SYSTEM\SALFAA26.ROM A000
100 PRINT "SALFAA V2.6 .... loaded at rom 2"
110 ?#B888=3
120 *LOAD C:\ATOM\SYSTEM\PCBOX.ROM A000
130 PRINT "PC-BOX ..... loaded at rom 3"
140 ?#B888=4
150 *LOAD C:\ATOM\SYSTEM\GAGS24.ROM A000
160 PRINT "GAGS V2.4 ..... loaded at rom 4"
170 ?#B888=5
180 *LOAD C:\ATOM\SYSTEM\JOSBOX.ROM A000
190 PRINT "JOSBOX ..... loaded at rom 5"
200 ?#B888=6
210 *LOAD C:\ATOM\SYSTEM\PCUTIL.ROM A000
220 PRINT "PC-UTILITY BOX . loaded at rom 6"
230 *LOAD C:\ATOM\SYSTEM\SOSPCWT.ROM 1000
240 PRINT "BRANQUART SOS .. loaded at #1000"
260 PRINT "Operating System loaded in rom !"
270 ?#4F7=0;?#4FA=0:LINK #FF3F

```


Color Graphics Monitor aansluiten op Electron/BBC

door roland leurs

Het zal de meesten onder ons wel overkomen. Vroeger waren we blij met een monochroom monitor of kleuren TV die op de Electron aangesloten kon worden. Helemaal geweldig was het als we die TV via de SCART plug konden aansluiten. Lekker gaaf beeld.

In ons PC zat een CGA kaart met daaraan een CGA monitor aangesloten. Wat een luxe! Vier kleuren in de laagste grafische mode. (Huh, de Electron was dus beter want die had er acht!).

Ach ... dat waren nog eens tijden. Maar ja, dat is nu voorbij. Tegenwoordig is 16 miljoen kleuren de normaalste zaak van de wereld. De Atom-in-PC heeft voor het eenvoudige rad van fortuin al minimaal EGA nodig. Waar gaat dat heen?

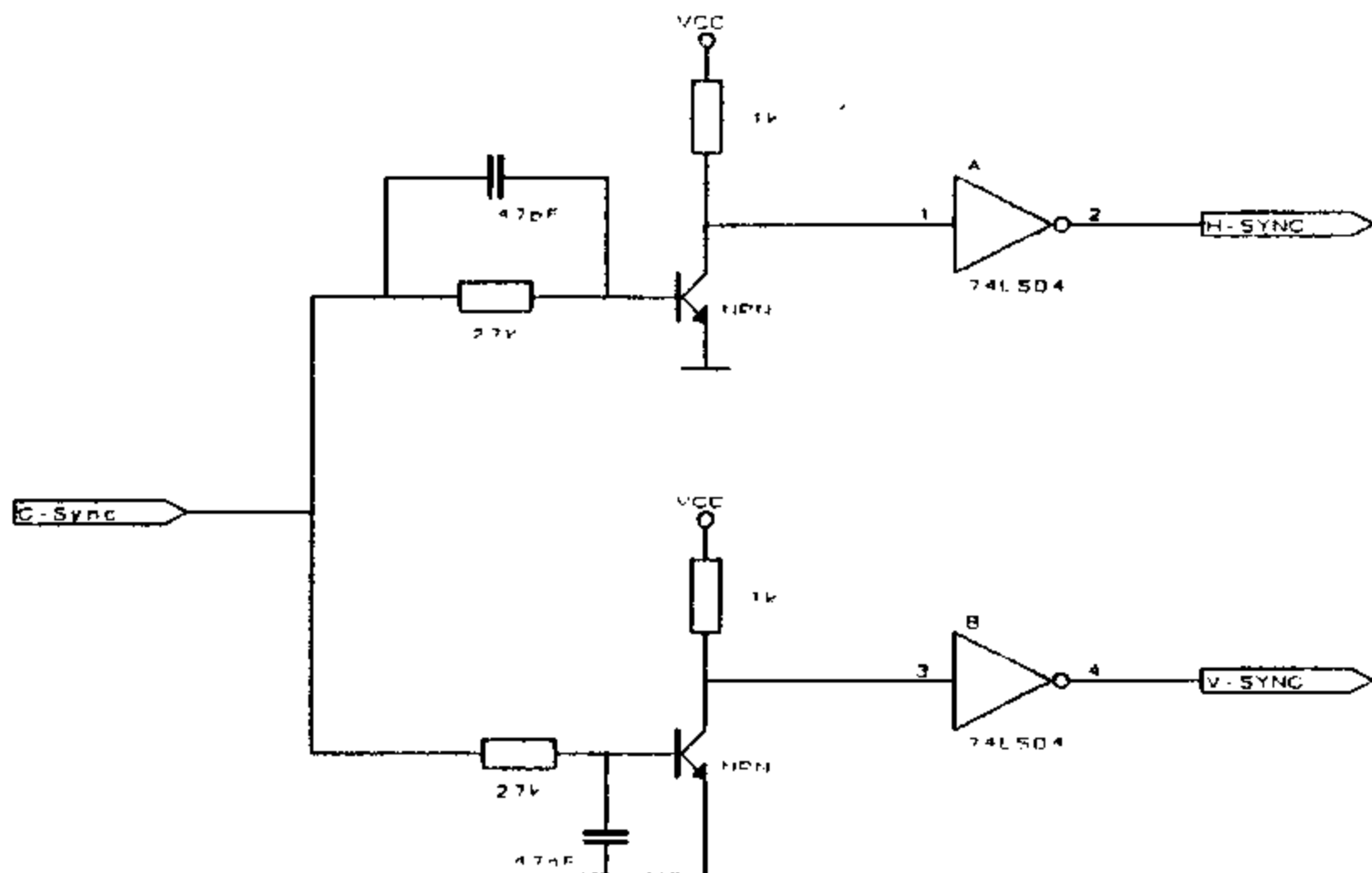
Waar dat heen gaat met de PC heeft verder niets met dit verhaal te maken. Maar we kunnen die CGA monitor toch nog goed gebruiken voor [de titel geeft het al aan] de Electron of BBC computer.

De uitgang van de Electron is bijna CGA, er komt immers Rood, Groen en Blauw uit. Dat is simpel aan te sluiten. Dan hebben we nog de synchronisatie. De Electron heeft een gecombineerd signaal voor horizontaal en vertikaal. Dat moeten we voor de CGA monitor even splitsen want die heeft twee synchronisatie ingangen.

Hoe pakken we dat aan? Ik begon met onze grote club-video-expert-deskundige Sjaak Geene te bellen. Als je zijn verhalen leest over de Atom kleurenkaart dan kan het niet anders zo zijn dan dat hij er iets voor heeft. En jawel. Twee dagen na mijn telefoontje had ik het schema in de brievenbus liggen. Het stelt niet veel voor. De gehele schakeling is te bouwen op een klein stukje experimenteerprint. Het schema staat op de volgende bladzijde getekend zodat niet iedereen naar Sjaak hoeft te bellen.

De voeding kunnen we uit de Electron zelf halen. Op de 6 polige DIN connector zit gelukkig een +5V aansluiting. Het intensiteit signaal heb ik via een weerstand van 4k7 aan de +5V gelegd. Daardoor zijn de kleuren iets feller en de kleur geel wordt dan ook als geel weergegeven. Als intensiteit aan massa ligt dan zijn de kleuren minder fel en geel wordt dan als een soort oranje/bruin achtige kleur weergegeven.

Rest mij tot slot om Sjaak even te bedanken voor het schema, met de vriendelijke groeten van Roland Leurs. :-)



Aansluitgegevens Electron (6 polige DIN connector):

pen 1	Rood
pen 2	Groen
pen 3	Blauw
pen 4	Comp. Sync
pen 5	ground
pen 6	+5V

Aansluitgegevens CGA connector (9 polig sub-D):

pen 1	ground
pen 2	ground
pen 3	Rood
pen 4	Groen
pen 5	Blauw
pen 6	Intensiteit
pen 7	niet gebruikt
pen 8	H-sync
pen 9	V-sync

Zullen we een lijntje trekken??

Nee, nee ik ga u geen voorstellen doen, die de narcoticabrigade brengt tot het fronsen van de wenkbrauwen. Nee, hier de beloofde aflevering over het tekenen van lijnen in Assembler. Gezien de enthousiaste opmerkingen over het vorige artikel over het tekenen van een puntje, heeft u wellicht al bij de brievenbus zitten wachten op wat komen gaat.

De bedoeling van dit artikel is, om u inzicht te geven in hoe een line-draw algoritme werkt. Het hier gekozen algoritme staat ook toe dat een ieder die met succes zijn Atom aan kan zetten, dit ook kan begrijpen.

1 Horizontale en verticale lijnen

Voor het schrijven van een programma dat uitsluitend rechte lijnen tekent, is niet echt veel nodig. Als voorbeeld een rasterwerkje, iets wat op een schaakbord lijkt. En u ziet: het trekken van lijnen is verbluffend eenvoudig.

```
10 rem schaakboard?
20 clear 0
30 for x = 0 to 190 step 10
40 for y = 0 to 190
50 move x,y; draw x,y
60 next
70 next
80 for y=0 to 190 step 10
90 for x=0 to 190
100 move x,y; draw x,y
110 next
120 next
999 end
```

Niets aan, zult u zeggen. En inderdaad: zolang er sprake is van een rechte lijn, dan is er niets aan de hand. Maar zo simpel als het trekken van rechte lijnen is, zo complex is het trekken van lijnen onder een willekeurige hoek.

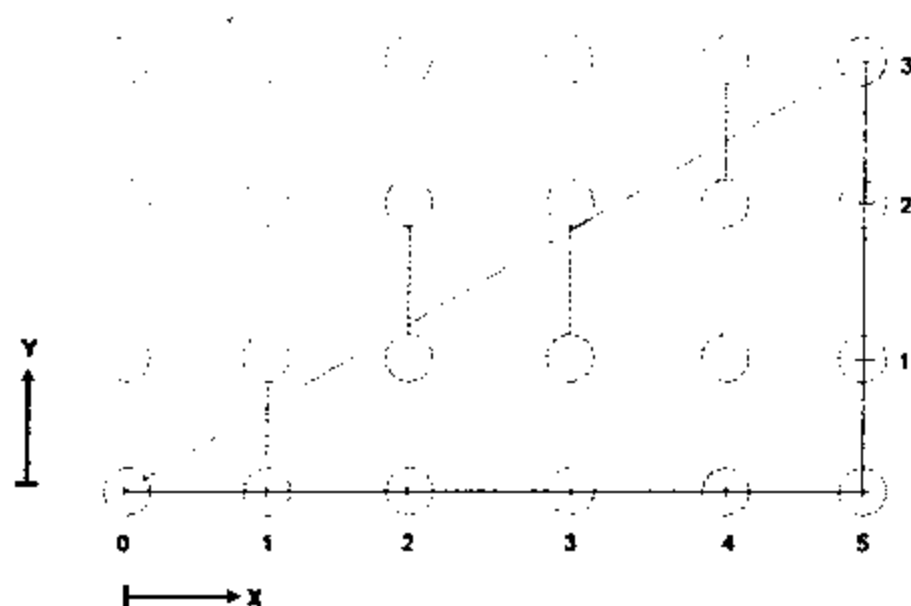
2 Diagonale lijnen

Nu komt het leukere deel aan de orde: het tekenen van lijnen welke niet horizontaal danwel verticaal zijn. Stel dat we een

lijn willen tekenen van 3 punten verticaal en 5 horizontaal. Gemakshalve wordt de Atom notering en presentatie aangehouden.

Zou een dergelijke lijn worden getrokken, dan raakt de lijn exact 0,0 en 5,3. De overige punten worden net niet helemaal gehaald. Nevenstaand figuur maakt duidelijk wat ik bedoel.

De grote vraag is: hoe maak je een verstandige keuze voor het kiezen van die punten zodat het geheel op een lijn gaat lijken. De gestreepte lijnen in het figuur pogen die lokaties aan te geven.



De gestreepte lijnen in het figuur pogen die lokaties aan te geven.

2.1 Het vermenigvuldigen met de factor

In een grijs verleden hebben we allemaal op school eens geleerd dat een lijn wiskundig gezien kan worden voorgesteld door de formule:

$$y = ax + b$$

A is hier het verhoudingsgetal, in ons voorbeeld $3/5$, en b is de constante, of offset zo u wil, voor als het feest niet op 0,0 begint. Met andere woorden: in dit geval kan elke y coördinaat worden berekend met $y = 3/5 * x$.

In dit geval is dat dus uiterst eenvoudig. We beginnen met $X=0$, doen dat totdat X gelijk is aan 5. Voor de Y coördinaat ligt dat wat moeilijker, want dat wordt een gebroken getal:

$$y = 3/5 * 1 \text{ geeft } 3/5$$

En voor $X = 2$:

$$y = 3/5 * 2 \text{ geeft } 6/5$$

Als de formule op deze wijze gebruikt wordt, geeft Y een indicatie waar het punten gezet dient te worden. Maar omdat punten alleen op een hele coördinaat getekend kunnen worden, zal er moeten worden afgerond. Onderstaande afbeelding maakt dat duidelijk.

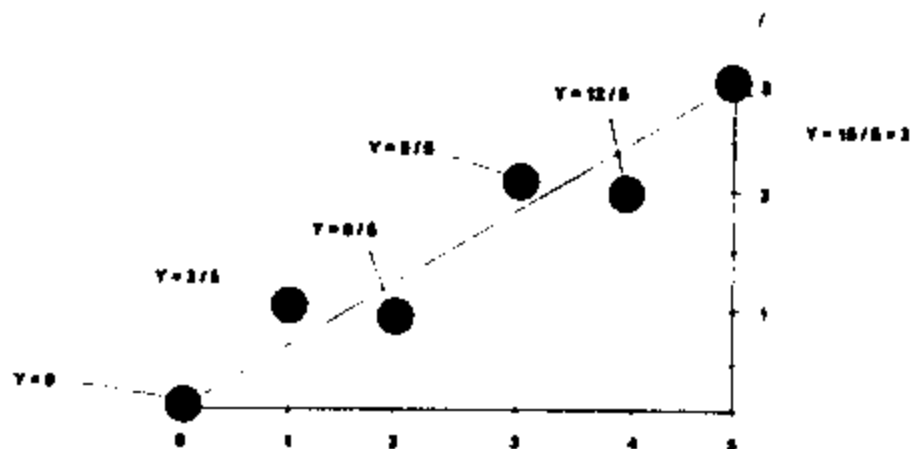
Maar wat is er mis met deze aanpak. Niets. Edoch... hebt u al een deelinstructie in de 6502 ontdekt, of wel eens gewerkt met getallen achter de komma in 6502 assembly?

Een ander belangrijk punt, is dat een aanpak op deze manier niet snel is. In dit voorbeeld gaat het om 6 punten. Maar

wanneer het gaat over honderden punten, dan wordt het zo traag als de kabinetsformatie. Dus snelheid is geboden.

Er zijn twee redenen voor waarom het vermenigvuldigen met een factor traag is. Ten eerste: de 6502 heeft geen vermenigvuldiging instructies. Zou het ze wel hebben, dan nog is dat te traag, omdat ieder punt individueel berekend moet worden.

Ten tweede: de computer moet met gebroken getallen moet rekenen, iets wat niet van nature in een processor is ingebakken. Het is ook mogelijk om met binaire fracties te werken, maar ook dat biedt weinig soelaas. Wat er nodig is, is volkomen andere aanpak:



3 Het Bresenham's Algoritme

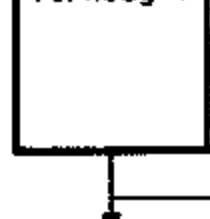
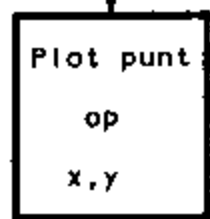
Hier komt het oude Bresenham algoritme om de hoek kijken, dat een methode biedt voor het trekken van lijnen. Dit algoritme is erg populair in de computer industrie: vrijwel elke fabrikant heeft een implementatie van het algoritme.

Kenmerkend voor deze methode is dat er een waarde, genaamd ErrorTerm, wordt bijgehouden die gerelateerd is aan het verschil tussen het punt waar het nieuw te plotten punt zou moeten komen en waar het punt terecht moet komen, omdat het nou eenmaal op een hele coördinaat moet komen.

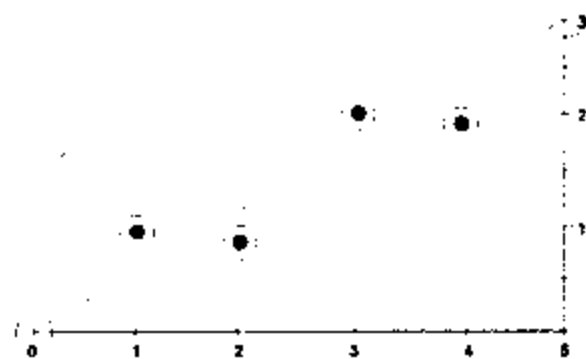
Iedere keer als de X waarde verhoogd wordt, wordt er 'iets' bij deze ErrorTerm opgeteld. Als die waarde van ErrorTerm nou maar groot genoeg wordt, dan wordt op, gegeven moment Y verhoogd en de waarde van ErrorTerm weer op een andere waarde gezet, door er een andere constante van af te trekken.

Bent u daar nog?? Ik geef toe het klinkt wat verwarrend, maar een plaatje doet weer wonderen:

Laten we wat zaken vaststellen: Weer een lijn tussen 0,0 en 5,3. Het verschil tussen begin- en eindpunt leggen we vast in Δ_x en Δ_y . Delta staat voor verschil. Verder moeten we de genoemde constanten vaststellen. Voor de X is dat HalfX , wat, u raadt het al, de helft is van Δ_x . Het gebroken gedeelte van deze deling is niet interessant, althans voor het functioneren van het algoritme, dus of je nou 13 of 12 door 2 deelt: in bieder gevallen leeft het algoritme met de waarde 6. In bovenstaand figuur is per coördinaat alles aangegeven.



Laat ons zien wat er gebeurd. X en Y starten op 0, net als ErrorTerm. Genoemd is dat Delta_X en Delta_Y respectievelijk 5 en 3 zijn. HalfX is 2. In bovenstaand figuur is duidelijk te zien wat er met



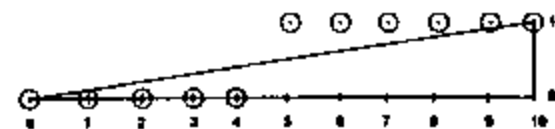
ERRORTERM	0	-2	1	-1	2
Add 3 (DELTA_Y)	3	1	4	2	
Result <= 2 ?	No	Yes	No	Yes	No
Subtract 5 (DELTA_X)	-2		-1		0
Increment Y ?	Yes	No	Yes	No	Yes

ErrorTerm gebeurd. Probeer aan de hand van het nevenstaande stroomschema het geheel te doorgronden.

Het begint op 0,0. X wordt met 1 verhoogd. Om nu te bepalen wat Y moet worden, wordt eerst Delta_Y (3) bij ErrorTerm opgeteld. Als het resultaat kleiner is dan de HalfX waarde (2), dan is er niets aan het handje in het andere geval wordt Y met 1 verhoogd en wordt de waarde Delta_X, die 5 is, van ErrorTerm afgetrokken. Dit circus herhaalt zich totdat de eindwaarde van X bereikt is.

3.1 Maak de stappen symmetrisch

Een van de vereisten aan een goed lijn algoritme, is dat de stappen symmetrisch dienen te zijn. Dat is met name van belang voor lijnen die bijna horizontaal of bijna verticaal. Immers daar is slechts een klein verschil in de coördinaten. Stel dat we een lijn willen trekken van 0,0 naar 10,1. Dus in JBF taal: de lijn gaat 10 stapjes naar rechts terwijl hij maar 1 stapje omhoog gaat. Op een bepaald punt moet de lijn dus overgaan van Y=0 naar Y=1. In nevenstaand figuur is dat verder



ERRORTERM	0	1	2	3	4	4	3	2	1	0
Add 1	1	2	3	4	5	4	3	2	1	0
Result <= 5 ?	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Subtract 10					-4					
Increment Y ?	No	No	No	No	Yes	No	No	No	No	No

uitgewerkt op basis van het Bresenham algoritme en u ziet: keurig op de 5 wordt de waarde van ErrorTerm overschreden en dus de Y coördinaat verhoogd.

4 Het lijnenspel

In onderstaande listing treft u een voorbeeld aan van het trekken van lijnen. Korthedshalve verwijs ik na het commentaar in de listing.

```

10 REM Tekenen van lijnen
11 REM #52/#53 lengte van de te plotten x as
12 REM #54/#55 lengte van de te plotten y as
13 REM #56 X-as positief of negatief
14 REM #58 Y-as positief of negatief
15 REM #57/#59 errorterm
16 REM #5a/#5b plot coördinaat x-as
17 REM #5c/#5d plot coördinaat y-as
18 REM zie blz. 35/36 atomware deel 3
20 DIM LL30
30 FOR N=0 TO 30; LLN=-1; NEXT
40 FOR N=0 TO 1
50 P=#7000; [
60 :LL30 LDX @#2           X als index register
70 :LL14 SEC              zet carry voor het aftrekken
80 LDA #5A,X             laadt lage byte nieuwe coördinaat
90 SBC #52,X             trek lage byte oude coördinaat er af
100 LDY #52,X            laadt lage byte oude coördinaat
110 STY #5A,X            zet lage byte oude coördinaat als eerste plot coo
120 STA #52,X            zet lengte lage byte van te plotten coördinaat
130 LDY #53,X            laadt lage byte coördinaat
140 LDA #5B,X            laadt lage byte nieuwe coördinaat
150 SBC #53,X            trek lage byte oude coördinaat er van af
160 STY #5B,X            zet lage byte
170 STA #53,X            zet lengte lage byte van te plotten coord.
180 STA #56,X            zet lengte hoge byte (positief of negatief)
190 BPL LL13             als positief dan wegwezen
200 LDA @#00             reset accu
210 SEC                  zet carry wat we gaan aftrekkn
220 SBC #52,X            trek negatieve waarde ervan af
230 STA #52,X            zet 'm terug
240 LDA @#00             reset accu opnieuw
250 SBC #53,X            hoge gedeelte van de aftrekbewerking
260 STA #53,X            en storen maar
270 :LL13 DEX
280 DEX                  verlaag index
290 BPL LL14             indien xas nog niet gedaan dan nogmaals
300 \ Bereken langste gedeelte van de lijn
310 :LL12 LDA #54        laadt lge byte van y coördinaat
320 CMP #52              vergelijk die met die van de x coördinaat
330 LDA #55
340 SBC #53              trek van hoge byte de x af
345 BCC LL7              spring indien x groter is
350 \ Y coördinaten zijn de grootste plot een lijn
360 :LL11 LDA @#00       reset de accu
370 SBC #54              trek lage byte Y coördinaat er af
380 STA #57              plaats waarde in pointer
390 LDA @#00             reset accu
400 SBC #55              trek hoge byte Y-coördinaat er af
410 SEC                  zet carry voor deling
420 ROR A                deel deze koter door 2
430 STA #59              plaats in half-y

```

440 ROR #57	deel laagste gedeelte door 2
450 :LL10 JSR #F678	romroutine voor plot punt
460 LDA #5C	
470 CMP #03C3	vergelijk met eindwaarde
480 BNE LL8	indien niet, dan verder
490 LDA #5D	
500 CMP #03C4	vergelijk met eindwaarde
510 BNE LL8	
520 :LL9 RTS	klaar is leen
530 :LL8 JSR LL3	pas errorterm aan
540 LDA #59	
550 BMI LL10	
560 JSR LL4	trek lengte van y coördinaat af en pas x aan
570 JMP LL11	plot de coördinaten
580 :LL7 \plot het geheel voor als x de grootste is	
590 LDA #53	laadt hoge byte van x coördinaat
600 LSR A	deel het door 2
610 STA #59	zet in errorterm
620 LDA #52 laadt lage byte x coördinaat	
630 ROR A	deel door 2
635 STA #57	zet resultaat terug
640 :LL6 JSR #F678	
650 LDA #5A	laadt lage byte geplotte x coördinaat
660 CMP #03C1	vergelijk met eindwaarde
670 BNE LL5	indien niet het einde, dan verder
680 LDA #5B	laadt hoge byte geplotte x coördinaat
690 CMP #03C2	vergelijk met eindwaarde
700 BEQ LL8	indien gelijk dan einde routine
710 :LL5 JSR LL4	pas errorterm aan en x coördinaat
720 LDA #59	laadt hoge gedeelte errorterm
730 BPL LL6	indien positief dan niets aan te passen
740 JSR LL3	pas errorterm aan voor Y coördinaat
750 JMP LL6	
760 :LL4 \ trek lengte y coördinaat van pointe af, upd x	
770 SEC	zet carry voor het aftrekken
780 LDA #57	laadt lage byte van errorterm
790 SBC #54	trek lage byte lengte y ervan af
800 STA #57	zet resultaat in errorterm
810 LDA #59	laadt hoge byte pointer
820 SBC #55	trek lengte hoge byte ervan af
830 STA #59	zet resultaat in errorterm
840 LDX @#00	laadt index regeiste voor x coördinaat
850 BEQ LL2	jump always
860 :LL3 \ tel lengte x as bij pointer, upd. Y	
870 CLC	clear carry voor het optellen
880 LDA #57	laadt lage byte errorterm
890 ADC #52	tel lage byte x coördinaat op
900 STA #57	zet resultaat terug
910 LDA #59	laadt hoge byte errorterm
920 ADC #53	tel hoge byte x coördinaat er bij op
930 STA #59	zet resultaat terug
940 LDX @#02	laadt x register als index voor y coördin.
950 :LL2 \bepalen positief of negatief	
960 LDA #56,X	laadt testbyte positief of neg. plotten
970 BPL LL0	indien positief
980 LDA #5A,X	laadt lage byte coördinaat
990 BNE LL1	test op verhogen hoge byte
1000 DEC #5B,X	verlaag hoge byte coördinaat
1010 :LL1 DEC #5A,X	verlaag lage byte coördinaat
1020 RTS	


```
1030 :LLO \ aanpassen coord. voor positief plotten
1040 INC #5A,X          verhoog lage coördinaat
1050 BNE LL8           test op verhogen hoge byte
1060 INC #5B,X          verhoog hoge coördinaat
1070 RTS               einde routine
8999 ;]
9000 NEXT
9999 END
```

Leuk een aardig, maar er zitten toch een paar puntjes in die wat aandacht behoeven.

4.1 Moeilijkheid 1

De eerste complexiteit is dat het X eindpunt van de lijn kleiner is dan het begin X punt, of het Y eindpunt kleiner dan het Y beginpunt. In die gevallen dient in plaats van het verhogen van X of Y een verlaging plaats te vinden. Kijk maar eens in de listing hoe dit gerealiseerd is.

4.2 Moeilijkheid 2

De tweede moeilijkheid is als de hoek tussen x en y meer of minder is dan 1, wat overeenkomt met 45 graden. Tot nu toe zijn we er vanuit gegaan dat deze hoek altijd minder is dan 45 graden. Is dit niet zo, dan 'groeit' de Y coördinaat sneller dan de X coördinaat. Met andere woorden: er zijn meer Y coördinaten dan X coördinaten. Een lijn tussen 0,0 en 3,10 moet dus anders behandeld worden dan die tussen 0,0 en 10,3. Ga dit zelf maar eens na. In de listing zijn twee stukken programma-tuur opgenomen: een voor een hoek kleiner of gelijk aan 45 graden en, hoe is het toch mogelijk, een voor groter dan 45 graden.

5 Uitbreidingen

Natuurlijk is het zo dat er verbeterde uitvoering, snellere algoritmen zijn etc. Het blijft nou eenmaal mensenwerk. Te denken valt aan zelfmodificerende coding, wat in ieder geval wat snelheidswinst oplevert. Dit is uiteraard wel een drama als er ge'debuged' moet worden of als het geheel in een eeprom wordt gestopt. Snellere algoritmen zijn er ook. Deze zijn echter beduidend complexer. Wie zich geroepen voelt: klim in de printer en meldt het in de Atom Nieuws!

INTIKKEN EN RUNNEN MAAR

KENT U ZE NOG, DIE KORTE PROGRAMMA'S DIE ZONDER UITGEBREIDE TOELICHTING IN ATOM NIEUWS STONDEN. STUUR UW KORTE, HANDIGE OF GEWOON LEUKE PROGRAMMA'S IN. DE LEUKSTE INZENDING VOOR DE VOLGENDE KEER WORDT BELOOND MET EEN LEGE DISKETTE !

```

10 PROGRAM BASIC PRINTER LIST
20 DIM N(32),R(255)
30 INPUT "FILENAAM "$N
40 F=FIN $N;IF F=0 THEN PRINT $7'"FILE NIET GEVONDEN"';END
50 C=BGET F
60 IF C<>#0D THEN PRINT $7'"BESTAND IS GEEN BASIC
PROGRAMMA"';GOTO E
70 INPUT "AANTAL REGELS PER PAGINA "P
80 INPUT "AANTAL KOPREGELS "K
90 @=5;X=0
100 PRINT "'LISTING WORDT AFGEDRUKT , , , ",$21$2
110 FOR X=K TO 1 STEP -1;PRINT ' ;NEXT X
120 C=BGET F
130 WHILE C<>#FF
140 C=C*256+BGET F
150 PRINT C
160 SGET F,R
170 I=0;T=0
180 WHILE I?R <> #0D
190 PRINT $I?R
200 I=I+1 ; T=T+1
210 IF T=75 THEN PRINT "" ";T=0;X=X+1
220 IF X=P THEN PRINT $12;FOR X=K TO 1 STEP -1;PRINT
';NEXT X
230 WEND
240 PRINT '
250 X=X+1
260 IF X=P THEN PRINT $12;FOR X=K TO 1 STEP -1;PRINT
';NEXT X
270 C=BGET F
280 WEND
290 ESHUT F
300 PRINT $12$3$6'
310 END

```

```

10 REM BIG PLANET
20 REM CREATIVE ATOM GRAPHICS
30 REM DOOR ROLAND LEURS
40
50 CLEAR 53 ; REM VOOR ET-4000 COMPATIBLE VGA KAART
60 COLOUR 48
70 A=400;B=300
80 S=300
90 K=S*S+1
100 FOR Y=-S TO S
110 X=INT(SQR(K-Y*Y))
120 Z=2*X
130 FOR I=-X TO X
140 L=0;IF A.R.INT(Z-X<I THEN L=2
150 PLOT(15-L),(I+A),(Y+B)
160 NEXT;NEXT
170 END

```

```

10 REM ONE PEG TO MANY ?
20 CLEAR 9
30 DIM XX(3),YY(3)
40 VCC 23,230,228,27,18,18,54,36,36,109,23,231,147,111,76,
76,220,152,152,184
50 VCC 23,232,192,224,48,48,56,24,24,28,23,233,73,219,219,
219,219,219,73
60 VCC 23,234,48,112,112,112,112,112,112,48,23,235,12,12,
12,12,12,12,12
70 VCC 23,236,109,36,36,54,18,18,27,228,23,237,184,152,
152,220,76,76,111,147
80 VCC 23,238,28,24,24,56,48,48,224,192
90 MOVE 232,64;DRAW 32,64;DRAW24,87;MOVE232,87;DRAW24,87;
DRAW56,151
100 DRAW 264,151;MOVE 264,128;DRAW72,128;DRAW56,96;DRAW248
,96;MOVE71,96
110 DRAW80,119;DRAW72,127;MOVE248,119;DRAW80,119
120 XX1=29;YY1=8;XX2=31;YY2=12;XX3=33;YY3=16
130 FORN=1TO3;VCC 31,XXN,YYN
140 VCC230,231,232,10,8,8,8,233,234,235,10,8,8,8,236,
237,238
150 NEXTN;VCC31,6,2;PRINT "SLECHT CAD PROGRAMMA (?) "
160 LINK #FFE3;TXMOD;END

```

REGIO-ADRESSEN.

Wilt U lid worden van de ATOM COMPUTER CLUB ?.
Neem dan contact op met de penningmeester van de regio waar U bij
ingedeeld wenst te worden. Deze kan U inlichten omtrent het
lidmaatschap.

Regio NOORD-HOLLAND :
P.v.Kuik, Zuideinde 54-a, 1843 JP Groot Schermer.
tel. 02997- 1902.

Regio DEN HAAG +ARNHEM
Th.Waayer, H.v.Boeijenlaan 66, 2273 DC Voorburg.
tel. 070-3862504.

Regio BRABANT-OOST + ZEELAND
J.Teulings, K.Doormansstraat 54, 5224 GL Den Bosch.
tel. 073-212888.

Regio LIMBURG + BELGIE
C.Rutkowski, Mgr.Buckstr.8 6121 KV Urmond
tel. 04498- 60136

Leden die behoren tot opgeheven regio's, danwel regio's die conform
de statuten geen lid meer zijn van de federatie, worden in verband
met de financ. administratie en de verzending van ATOM-NIEUWS, door
de federatie toegewezen aan de nabije regio's.
Zo men tegen deze indeling bezwaar heeft, om welke reden dan ook,
kan men de regio van eigen keuze opgeven aan de penningmeester van
de federatie: T.Rutten , zie de pagina van de federatie in dit
blad.

Bij het aangaan van het lidmaatschap kunt U de contributie over-
maken op de rekening van de federatie. Vermeld hierbij uw volledige
naam, en adres , alsmede evt. de regio waarbij U wenst te worden
ingedeeld.